CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B1
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

JAN 21 1981

Item 830-H-10

NAS 1.55:2166

COMPLETED

NASA Conference Publication 2166

ORIGINAL

Numerical Grid Generation Techniques

Proceedings of a workshop held at
NASA Langley Research Center
Hampton, Virginia
October 6-7, 1980

NASA

NASA Conference Publication 2166

Numerical Grid Generation Techniques

Proceedings of a workshop sponsored by the
National Aeronautics and Space Administration
and the Institute for Computer Applications
in Science and Engineering and held at
NASA Langley Research Center
Hampton, Virginia
October 6-7, 1980

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1980

j

PREFACE

This document contains a set of papers presented at the Workshop on Numerical Grid Generation Techniques for Partial Differential Equations held at Langley Research Center October 6-7, 1980. The workshop was organized to assess the "state of the art" in grid generation and to assemble the individuals most involved with the technology to exchange ideas and establish prospects for the advancement of the technology.

The workshop was divided into three primary categories: classical techniques (complex variables); differential-systems techniques; and algebraic techniques. Intermixed in these categories were papers on adaptive grid generation and the analysis of errors caused by grids in the solution of partial differential equations. Herein, the three invited papers are presented first and are followed by the other papers in alphabetical order according to the first author's name.

It is apparent from the papers that two-dimensional grid generation is highly advanced. Complex-variable techniques, differential-systems techniques, and algebraic techniques are demonstrated to be viable for a wide variety of two-dimensional problems with complex boundaries and topologies. In some cases, associated computer programs are available for general distribution.

Progress is being made in the areas of adaptive grid generation and the analysis of how grids affect the solution of partial differential equations. However, only relatively simple problems have been considered thus far, and further work needs to be done in this area.

Several papers concerning three-dimensional grid generation were presented. However, the present "state of the art" for this area is highly restrictive. The construction of arbitrary three-dimensional grids needs to be done prior to the solution of many partial differential equation systems of practical interest, and considerably more work needs to be done in this area.
Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

Robert E. Smith

BLANK PAGE

BLANK PAGE

CONTENTS

PREFACE iii

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS 241
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES 295
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS 307
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES 329
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS 331
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW 377
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING UNIFORMLY SPACED COMPUTATIONAL MESH 385
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545
J. C. Wu and U. Gulcat

ATTENDEES 561

GRID GENERATION USING CLASSICAL TECHNIQUES

Gino Moretti

Polytechnic Institute of New York
Farmingdale, N. Y. 11735

1. Historical

Conformal mapping has been used as a tool for solving problems in fluid mechanics and electromagnetism for more than one hundred years. Riemann's (somewhat incomplete) proof of the possibility of mapping closed contours on circles dates from 1850 [1]. Schwarz introduced what is now known as the Schwarz-Christoffel transformation in 1869 [2]. In the same year, Kirchhoff and Helmholtz used conformal mapping to solve classical problems of flows with free surfaces [3]. The importance of conformal mapping in fluid mechanics in the second half of the nineteenth century and the first quarter of the present one stems from the well-known property of invariance of the Laplace equation through conformal mapping; the mapping itself is, indeed, defined by a Laplace equation. Since the theory of images allows any incompressible, potential flow defined by given singularities to be determined in the presence of a circle or a straight line, the analogous problem in the presence of an arbitrarily given contour is reduced to the problem of mapping the contour onto a circle or a straight line. In so doing, the problem of determining the analytic function representing the complex flow potential is split into two parts. One consists of finding the potential for the circle, and that is easy. The other still requires the determination of an analytic function to define the mapping, which is as difficult a task as the original problem [4, page 77]. Nevertheless, it is much more appealing to people who have a visually oriented mind, and it can be subdivided in a number of successive steps, each one of which is easy to understand geometrically.

The Joukowski mapping,

$$z = \zeta + \frac{1}{\zeta} \qquad (1)$$

discovered in 1910 [5], is extremely simple and easy to handle. Using a circle as the basic contour in the ζ-plane, it can generate a two-parameter family of cusped profiles and a three-parameter family of smooth contours ranging from familiar ellipses to peanut-shaped figures (Fig. 1). Despite their poor aerodynamical properties, the Joukowski profiles played a crucial role in the understanding of the mechanism of lift and, consequently, in the establishment of the theory of flight as a science. In creating his profiles, Joukowski introduced an idea which turned out to be extremely rich in consequences: that is, that a shifting of the center of the circle is sufficient to produce an airfoil having either camber or thickness or a combination of both.

Another major contribution dates from 1918 when von Kármán and Trefftz [6] observed that, if (1) is recast in the form:

$$\frac{z-2}{z+2} = \left(\frac{\zeta-1}{\zeta+1}\right)^2 \tag{2}$$

it can be interpreted as a combination of two bilinear mappings and a power. If δ is used instead of 2 in (2), the Kármán-Trefftz mapping results:

$$\frac{z-\delta}{z+\delta} = \left(\frac{\zeta-1}{\zeta+1}\right)^\delta \tag{3}$$

which, for $\delta=2-\epsilon/\pi$, produces an airfoil with a finite internal angle, ϵ, at the trailing edge. The property of a power,

$$z = \zeta^\delta \tag{4}$$

to eliminate sharp corners, was known and had been applied since the times of Schwarz and Christoffel, but the importance of its application in (3) should not be underestimated.

The role played by the exponential (and, conversely, the logarithm) and by all functions directly related to it (direct and inverse trigonometric and hyperbolic functions) in generating infinite cascades can be traced back to König, 1922 [7]. Transformations of this kind, for example,

$$z = e^{i\theta} \log \frac{\zeta-\kappa}{\zeta+\kappa} + e^{-i\theta} \log \frac{\kappa\zeta-1}{\kappa\zeta+1} \tag{5}$$

have essential singularities at infinity. Therefore, the image of the physical point at infinity in the plane of the circle is generally a pair of spiral-vortices at a finite distance, a geometrical property related to the ability of a cascade to deflect the flow.

Finally, in 1933 Ferrari found the basic mapping for biplanes. He showed that a doubly periodic function was needed, that is, an elliptic function. His paper [8] is the first application of elliptic functions to wing theory. [8].

Important as these basic mappings were, they did not solve the problem of mapping an arbitrarily given contour onto a circle. They generate a closed contour without sharp corners which, in a favorable case, may look like a quasi-circular contour which reminds one of a potato (the word "potato" has actually been used occasionally in the literature to designate such figures and, although not defined in mathematical terms, seems to be as acceptable as "quasi-circle"). The technique for the mapping of the quasi-circle onto a circle, proposed by Theodorsen in 1931 [9], actually brought conformal mapping from the speculative level down to practical levels of aeronautical engineering. Its basic idea is indeed extremely simple; if the center of the circle and the centroid of the quasi-circle are located in the origins of their respective planes, and some scaling is used to make the areas of the two figures coincide, then the mapping can be expressed in the form:

$$z = \zeta \, e^{f(\zeta)} \qquad (6)$$

where $z = re^{i\phi}$, $\zeta = e^{i\theta}$ and the modulus of f is small. Therefore, two equations follow:

$$\ln r = A(\theta) \quad , \quad \phi - \theta = B(\theta) \qquad (7)$$

if $f = A + iB$. On the circle, A and B are conjugate Fourier series. If ln r were known as a function of θ, A could be determined by expanding ln r into a Fourier series, and B would follow automatically. Since ln r is known, instead, as a function of ϕ, iterations are necessary to make the two equations compatible. Alternatively, values of $\phi - \theta$ can be obtained rephrasing the problem as an integral equation, so that the formula to be solved by iteration is the Poisson integral:

$$\phi-\theta = \frac{1}{2\pi}\int_0^{2\pi} \ln r \cot\frac{\omega-\theta}{2}\, d\omega \tag{8}$$

Convergence of the procedure is assured if the potato is "star-shaped", that is, if its contour is crossed only once by any radius issued from the center [10]; therefore, the procedure is generally safe and efficient.

Applications of the Theodorsen method to airfoils, biplanes and cascades were pioneered by Garrick [11,12,13], who used the transformations introduced by Joukowski, Ferrari and König to produce quasi-circles as intermediate images of given contours.

For a better understanding of the rest of this paper, three points should be made.

1) Mapping of a contour into an exact circle was necessary, when problems of incompressible, potential flow had to be solved, in order to take full advantage of the invariance of the Laplace equation and of the easy evaluation of the flow past a circle.

2) Since the only values of interest belonged to the rigid contours, all calculations could be limited to the circumference of the circle.

3) Nevertheless, all calculations, with the exception of the ones related to the Joukowski mapping, were extremely cumbersome when performed by hand. Most of the time, the Joukowski mapping was preferred to the Kármán-Trefftz mapping. Cascade and biplane problems were rather analyzed by approximate methods. The Hilbert integral, with the entire circumference subdivided in just a dozen intervals, was preferred to the more laborious Fourier expansions.

Such practical difficulties and the shifting of interest towards problems in compressible flow contributed to relegate conformal mapping into the background in the forties. When computers came about, there was not so much a demand for solution of incompressible, potential flow problems, and the consequent application of conformal mapping techniques.

2. Conformal mapping as a grid generator

Nevertheless, in recent times conformal mapping has again been looked upon in connection with problems of compressible flow, but of course the use of it as a device to solve the same equations of motion in a simpler plane has been dropped, since neither the Euler equations nor the Navier-Stokes equations are invariant by conformal mapping. The reason for revival stems from the fact that in contemporary numerical gas dynamics the equations of motion must be solved in the entire flow field, because of compressibility effects, viscous effects, presence of vorticity, etc. Contrariwise, for incompressible, potential flow it is theoretically correct and practically convenient to search for the solution on the rigid contour only. A computational grid is necessary, upon which the equations are to be discretized. When two space-like variables are involved, conformal mapping is a very convenient tool to generate a computational grid.

If the boundaries of the flow field are mapped onto a circle centered at the origin of a complex plane $\zeta = \rho e^{i\theta}$, the network of ρ=constant and θ=constant lines is conversely mapped onto the physical plane as an orthogonal grid, well draped around the rigid contours because the latter are the image of the circle itself.

Since the grid is orthogonal, the vector operators in the physical plane can be expressed directly and easily in terms of their counterparts in the mapped plane.

The search for an appropriate mapping is actually a search for a single function of a complex variable, a task by far simpler than the search for two functions of two independent variables. All the findings of more than a century, properly digested and interpreted as briefly shown in Section 1, can be put together to suggest the most appropriate mapping for a particular task. Our computers have eliminated the tediousness, inaccuracy and material errors which used to hamper the hand calculations of half a century ago; they also allow the necessary data for a very refined grid to be evaluated in a fraction of a second. Complex arithmetic in FORTRAN and other languages of the same level reduces coding to just about a rewriting of the basic equations (except when angles contained in more than one quadrant are involved).

Practical applications of conformal mappings to generate grids have been developed in the last decade following two separate lines. One proceeds from the observation that any conformal mapping is defined by an analytic function, and that real and imaginary parts of the latter are harmonic functions. Therefore, the conformal mapping problem consists of solving two Laplace equations. The task can be formulated in strictly numerical terms, using modern, fast Laplace solvers. This viewpoint can be traced back to 1923 [14] and has been made popular by a 1975 paper by Thompson et al. [15]. The other puts the emphasis on the use of closed form analytical expressions for the mapping functions. It seems that the first non-airfoil related application of the technique was presented in 1972 [16], although the same problem was reformulated in a more general form only in 1974 [17]. We will try to analyze here the philosophy of the approach and to show how it works in practical cases.

3. Philosophy of the 'closed form' approach

We will begin by discussing some of the advantages of a closed form approach. To fully appreciate them, let us focus our attention on what the majority of problems of current interest are: Two-dimensional, unsteady flows or three-dimensional, steady flows. In the former, the physical space is two-dimensional but the computational grid may be variable in time; in these cases we need a grid, generated by conformal mapping at every instant of time, but depending on parameters which are functions of time. In the latter, we may find it convenient to create a computational grid on planes defined by two of the three space coordinates, and again letting the grid change as a function of the third coordinate. To organize the following formulae, let us stipulate that the Cartesian coordinates on any plane to be mapped are x and y, and that a complex coordinate is defined,

$$z = x + i\,y \tag{9}$$

using the symbol t either for time or for the third space coordinate. The mapping of each z-plane onto another complex plane, defined by a variable ζ, will then be accomplished by an analytic function:

$$\zeta = \zeta(z;c) \tag{10}$$

where c is shorthand for any parameter, function of t.

As in every flow problem, it is not so much the coordinate of a point as the derivatives at that point which matter. First, we need the derivatives of the coordinates in the physical plane with respect to the coordinates in the mapped plane, and vice versa, contained in the complex expression:

$$g = \frac{d\zeta}{dz} \tag{11}$$

Then, we need second derivatives, which appear in dealing with the curl of the velocity vector, and these entail dg/dz. Finally, we need derivatives with respect to t, viz. $\partial\zeta/\partial t$ and $\partial g/\partial t$, which are computed from (10) and (11) by differentiating with respect to c and keeping z constant. Obviously, it is very convenient to have the mapping expressed in a closed analytical form since all derivatives are also expressed in closed form and can be exactly evaluated where and only where they are needed, that is, at the computational nodes. The argument is particularly interesting with regards to the t-derivatives. If the grid changes in t, the values of z generally change in t at each nodal point; therefore, numerical differentiation at a constant z may be cumbersome and inaccurate.

There is another case where having a closed form mapping is convenient. Potential (but not incompressible) flow problems, with the flow field extending to infinity, are commonly solved by relaxation. Commonly, the physical potential is expressed as the sum of an unknown and the potential of a flow, satisfying the proper physical conditions at infinity. The latter is easily formulated in terms of the transformed variables if the mapping is defined by a closed form function.

Then, a difficulty seems to arise. As we said in Section 1, in most cases it is not possible to find a closed form mapping which provides an exact circle as the image of a given contour; it is clear that Theodorsen's step is not a closed form function. Nevertheless, the difficulty is only apparent. Let us assume, indeed, that we know a function capable of transforming a given contour into a quasi-circular potato. Let us consider a computational plane with two variables, X and Y, relating X to ρ and Y to θ. The functions $X(\rho)$ and $Y(\theta)$ will be so defined that $X=0$, $Y=0$, $X=1$ and $Y=1$ on each boundary, in turn. The computational region in the (X, Y) plane is, thus, the interior of a square. In most cases, the functions $X(\rho)$ and $Y(\theta)$ are used to apply proper stretching of coordinates in either direction, in order to concentrate nodes where necessary. The computational grid is orthogonal and divided into equal intervals. The corresponding grid in the ζ-plane is not orthogonal (not only because the $X=0$ line is not an exact circle, but also because the other boundaries may not be circles or straight lines either). Consequently, the grid in the physical plane is not orthogonal. Accuracy, however, is not impaired for want of orthogonality. The equations, originally written in the (ρ,θ) frame, are recast using X and Y as independent variables. Some additional terms will appear; the very important boundary condition on the rigid surface will be properly written by stating that the normal velocity component (not just the ρ-component) vanishes. In conclusion, we are not expecting any dramatic advantage from having a circle as one of the boundaries, and therefore we may consider our mapping problem solved when we find a quasi-circle as the mapped image of the given boundary. The Theodorsen step can be dropped with all its additional burden of iterations, Fourier expansions, spline fittings, etc.

Let it be clearly stated that we are not afraid of unsurmountable difficulties or unaffordable computational times connected with the Theodorsen step. That old (but hard to execute by hand) way of computing Fourier expansions, which has been given a new popularity under the FFT label serves the purpose egregiously well. Dropping the Theodorsen step, whenever possible, is justified by our desire of achieving a solution to the problem in a closed form, with a view to the formal calculation of derivatives, particularly with respect to t. There is a definitive need for them in all problems where the grid depends on t.

This philosophy was clearly exposed in [16] and accepted by Jameson [18,19]. We will discuss some applications in Sections 5 through 8. Before that, we will mention some of the techniques which we tend to classify as numerical, rather than analytical.

4. Conformal mapping as a Neumann problem

The closed form approach may look like an empirical attempt to solve the problem, on the basis of analogies and imaginative variations, and therefore strongly dependent on the mental structure of the investigator. I have heard the word "art" used in a derogatory sense in connection with this type of work, on other occasions. Actually, good science is the product of personal ingenuity and crafty skill. What tends to be classified as "scientific" in these days is rather "technological", that is, some process which has been sealed in a black box for general purposes.

Whatever the consequences, there is no doubt that the quest for an organized conformal mapping procedure is legitimate. It is also classical. Green's formula:

$$\oint (u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n}) ds = 0 \qquad (12)$$

for two harmonic functions, u and v, regular in the domain surrounded by the closed contour on which the integral is made, and on the contour itself, dates from 1828. With any two points on the contour denoted by $\zeta = \rho e^{i\theta}$ and $z = re^{i\phi}$, and with $\zeta - z = \sigma e^{i\alpha}$, it follows from (12) that

$$u(z) = -\frac{1}{\pi} \oint [u(\zeta) \frac{\partial \ln \sigma}{\partial n} - \frac{\partial u}{\partial n} \ln \sigma] ds \qquad (13)$$

All formulae related to conformal mappings can be obtained from (13); $\partial u/\partial n$ is generally known from its conjugate, the variation of the tangent along the contour. Then (13) becomes an integral equation for u.

Many different forms of the equation can be obtained, if one makes use of well-known properties and formal rules, such as: the Cauchy-Riemann conditions, integration by parts, Schwarz and Poisson's integrals, integrals defining the coefficients of a Laurent series. Different forms are also obtained by taking the basic contour as a circle or as a straight line, and by defining u either as the logarithm of z or as the logarithm of g. For example, if we start from (6) and from a circle, the other contour being a quasi-circle, we obtain the Theodorsen mapping in its integral form (8); the Fourier series form follows easily. The same equation (6) and somewhat different integral equations have been used by Symm [20] and Hayes et al. [21] to produce numerical techniques which are not restricted to mappings of quasi-circles onto circles, as in Theodorsen's, but apparently can handle any (probably, star-shaped) contour. If we use the logarithm of g, we can interpret Theodorsen's ideas in terms of derivatives, á la Timman [22], a variation which seems to offer some numerical advantages [17]. If we start from a straight line and an arbitrary, closed contour, using again the logarithm of g, we obtain what Davis [23] presents as a generalization of the Schwarz-Christoffel mapping to a polygon with an infinite number of infinitesimal vertices:

$$\log g = \frac{1}{\pi} \int \log (\zeta - b)\, d\beta \qquad (14)$$

Anyone interested in these types of comparisons could profitably read a paper by Birkhoff et al. [24] which is outdated only from a computational viewpoint.

Proceeding in the opposite direction as Davis, the Schwarz-Christoffel formula for a polygon with a finite number of vertices can be found as a particular case of (14). This formula is, in principle, a very powerful mapping tool. It can map a polygon on a circle, without restricting number, location and aperture of vertices, or the lengths of the sides, and permitting vertices to be located at infinity. It is really a definition of g, rather than of ζ, which must be obtained by complex integration in a numerical form, in almost all the cases. This is not a major shortcoming, however, since numerical integration can be performed quickly and efficiently. The derivative, dg/dz is straightforward. Nevertheless, it is known that the coefficients, ζ_i and δ_i which appear in the formula

$$g = \Pi\ (\zeta - \zeta_i)^{\delta_i} \tag{12}$$

must be obtained by trial-and-error iterative processes (see, for example, [23,25,26]). If the grid does not depend on t, its coefficients can be determined once and for all. In this case the Schwarz-Christoffel formula belongs to the category which we consider in the present paper. It does not if the grid depends on t. The same may be said for all mappings obtained by solving a Neumann problem via iterations on an integral equation or Fourier expansions.

5. Kármán-Trefftz mapping for airplane cross sections

We will now consider some mappings using a finite number of closed form relations.

The numerical analysis of a steady, supersonic flow past an airplane may be performed by marching in an axial direction and updating values at successive cross-sectional planes. The region of interest in each plane is bounded by the section of the body and the section of the bow shock. Conformal mapping of the body onto a quasi-circular shape provides a grid which tends to become a polar grid at infinity and therefore is the best suited to adjust to the shape of the body, whatever it is, and to the almost circular shape of the bow shock. The body shows a number of edges and corners as those indicated by letters in Fig. 2. If the airplane is arrow-winged, stations will be reached where the body will be composed of three unconnected parts; if two fictitious lines are drawn between the fuselage and the trailing edges of the wings, again we can see corners and edges at all points denoted by letters in Fig. 3. Observing that the corners and edges always come in pairs, because of the symmetry of the cross-section, we can think of eliminating them by successive applications of the Kármán-Trefftz mapping (3), with the singular points either on the contour or, in case of rounded edges and corners, slightly inside. Note that edges require values of δ between 1 and 2 and corners require values of δ less than 1. When the latter are applied, the corner is open, but so is the rest of the plane, part of which may end up in a second Riemann sheet. In principle, this is not an obstacle to the removal of corners, because the portion of plane which disappeared will be recalled when removing the next edge. In practice, a quite cumbersome additional piece of logic must be added to identify points belonging to the second Riemann sheet. The trouble can be avoided by executing the mappings not according to the order of appearance of a corner or edge along the contour but in a sequence of decreasing values of δ.

From the viewpoint of coding, the repeated application of the Kármán-Trefftz procedure has many advantages:

1) Regardless of the number, position and aperture of corners and edges, the same operation is used, which means the code may be written in the form of a loop and applied as many times as necessary, automatically,

2) The mapping can be inverted, and the inverse mapping has the same form; therefore, the same routine can be used for the direct and the inverse mapping,

3) The derivatives are easily coded; for example g, as defined by (11), is actually the product of the derivatives of each intermediate step, and the logarithmic derivative of g is the sum of the logarithmic derivatives of such steps.

To show how close to a circle the image of a fuselage with two sections of arrow wings is, Fig. 4 presents a computational grid in the physical plane and its image in the mapped plane. Naturally, with the bow shock very close to the leading edge of the wings, its own image is far from a circular shape but, as we said above, departure from orthogonality of the grid is not jeopardizing accuracy, and this is particularly true in the vicinity of the bow shock, where the flow is uncomplicated. Details of the technique and its application to the arrow-winged airplane problem can be found in [17].

6. Imaginative devices

There are no limits to the number of shapes which can be obtained by executing elementary mappings in a sequence and using a little ingenuity. Indeed, bilinear transformations, powers and logarithms are the building blocks with which one should learn to play, always keeping in mind that rotations and translations can be cleverly used to locate singularities where needed. Here is an interesting example, due to Rossow [27], which is entirely expressible by a sequence of bilinear mappings and powers (Fig. 5).

A circle, centered at the origin, is translated upwards and then a Joukowski mapping changes it into an arc of a circle, counted twice. The arc is rotated about one of its ends and a new singular point is defined somewhere along its length. A new Joukowski mapping is applied, in reverse, so that the portion of the arc between the two singular points become a circle again, and the portion left outside remains appended like an infinitely thin tail. Finally the circle is relocated, and a third Joukowski mapping is used to transform it into a Joukowski profile; the little tail becomes a flap or spoiler, whose location and length can be controlled by changing the parameters used in the successive steps. According to our quasi-circular philosophy, this very simple mapping can be used for any airfoil with attached (but not infinitely thin) spoilers or flaps.

I faced a similar problem when confronted with generating a grid for the calculation of the precursor muzzle blast [28]. I needed a grid shaped as in Fig. 6; the contour defined by D'C'CD can be mapped onto the real axis of a w-plane, and its exterior onto the upper half w-plane by a simple Schwarz-Christoffel transformation [29, page 159], but this mapping would not provide a family of grid lines issuing from what has to be interpreted as the bore of the gun and wrap around the barrel, as in the figure. The problem was solved by defining two new singular points, B and B', in the w-plane and applying to this plane an inverse Joukowski mapping onto a ζ-plane. Radial straight lines and concentric circles in the ζ-plane are now producing the wanted grid in the z-plane. Note that the position of B and B', being arbitrary, permits the ratio of outer-to-inner radius of the barrel to be matched (Fig. 7). With the change of notation: $w = Z + 1/Z$, which simplifies the coding, the mapping is thus defined by:

$$z = (r_o/\pi)[(Z^2-1/Z^2)/2 = \log Z^2 - i\pi]$$

$$Z + 1/Z = 2 B (\zeta + 1/\zeta) \qquad (15)$$

In the problem just described, the computational region is limited, on one side, by the precursor shock which moves out in time. Therefore, the computational grid is a function of time, but the dependence on time shows only through the stretching parameters; the mapping itself remains invariable in time and the equations of motion carry no terms of the type $\partial g/\partial t$ or $\partial \zeta/\partial t$. When a mapping is needed for the same problem in the presence of a protruding bullet, however, the contour itself changes in time and so does the mapping. Once more, the problem can be solved with little additional effort. One can start with a half circle (Fig. 8), reduce it to a half ellipse using a Joukowski mapping, and then apply a Kármán-Trefftz formula to change the angles between the contour and the real axis. After that, the mapping continues as defined by (15). The axis ratio of the ellipse keeps growing as the bullet nose advances. The power in the Kármán-Trefftz mapping decreases from an initial 1 to a minimum value of 1/2. At this stage, the contour in the physical plane makes a 90 degree turn at B and B', as required to accomodate the grid to the side surface of the bullet (Fig. 9).

7. Different mappings for the same geometry

In solving problems of incompressible, potential flow there is no ambiguity about the choice of the mapping since the contours of interest are specified exactly and the mapping function has to be regular over the entire flow field (boundaries excluded). In the present context, though, different mappings may accomodate the same contours but generate completely different grids, as we have mentioned in the preceding Section. Care must be taken to use a mapping whose grid is the best suited for the problem in hand. This problem is currently acute in cascade analysis, as we will mention later; here I would like to show it in a simpler connection.

We want to generate a grid for the intake of which Fig. 10 shows the centerline and the shroud. We can drape our grid about a semi-infinite slit, parallel to the real axis in the upper half-plane. Two mapping functions, apparently opposed to one another, can serve the purpose. The first is:

$$z = \zeta + e^{\zeta} \tag{16}$$

This function maps the slotted half plane onto a strip, which is a very convenient domain for our computational variables X and Y. The corresponding grid (Fig. 11) is convenient for the interior of the intake, but it needs some stretching to provide resolution to the exterior of it; in particular, the outer surface of the shroud is poorly resolved. The other mapping is defined by:

$$z = \zeta + a \log \zeta \tag{17}$$

and it maps the same region of the physical plane onto the entire upper ζ-half-plane. Cartesian coordinates in the ζ-plane produce the grid of Fig. 12. In this case, the grid is very good outside and very poor inside. If we start from polar coordinates in the ζ-plane, we obtain the grid of Fig. 13. The general appearance of the grid lines recalls Fig. 11, but the situation is reversed: the resolution is very poor inside and very good outside; in this case, an accumulation of ρ=constant lines near the origin of the ζ-plane is necessary to generate some coordinate lines inside the intake. An application of (16), with a stretching of coordinates to relax the resolution inside the nacelle, has been made by Caughey and Jameson [30].

Anyway, these mappings are obtained in a very straightforward manner, and more complicated manipulations [31] do not seem necessary.

8. Biplanes, revisited

Ives [32] has proposed a technique for the mapping of two airfoils which is rich in possible consequences. We have seen in Section 5 how the successive application of Kármán-Trefftz mappings can eliminate corners and edges on a contour, producing a quasi-circular shape. A similar idea, exploiting the theory of images with respect to a circle, allows two airfoils to be mapped on two concentric circles. Two Theodorsen mappings are used as intermediate steps, as follows (Fig. 14). First, a Kármán-Trefftz formula is applied to transform one of the airfoils into a quasi-circle, and then the quasi-circle is transformed into an exact circle by the Theodorsen technique. At this stage, the second airfoil is still shaped as an airfoil although with a different shape. The next step manages to transform the second airfoil into a quasi-circle, _without distorting the first circle_. The problem is solved by using the product, side-by-side, of two Kármán-Trefftz formulae; the first contains the two singular points pertinent to the second airfoil (a and b) and their counterparts in the mapped plane (α and β); the second contains the images of such points ($1/a^{\bullet}, 1/b^{\bullet}$ and $r_0^2/\alpha^{\bullet}, r_0^2/\beta^{\bullet}$, respectively, where conjugates are denoted by $^{\bullet}$ and r_0 is the radius the image of the first circle in the mapped plane):

$$\frac{(\zeta-\alpha)(\zeta-r_0^2/\alpha^{\bullet})}{(\zeta-\beta)(\zeta-r_0^2/\beta^{\bullet})} = [\frac{(z-a)(z-1/a^{\bullet})}{(z-b)(z-1/b^{\bullet})}]^{\delta} \qquad (18)$$

Finally, the second quasi-circle is moved inside the first circle by a bilinear transformation in such a way that its centroid coincides with the center of the circle, and a second mapping of the Theodorsen type is applied. Again, such a mapping must take into account that the flow region is a ring between two concentric circles. Therefore, the exponent in (6) cannot be a simple Taylor series, as it would be if the flow occurred inside a circle, or a Taylor series of negative powers, as it would be if the flow occurred outside a circle, but it must be a two-sided Laurent series; we express this need by saying that the Theodorsen mapping is defined in this case by:

$$z = \zeta \, e^{f(\zeta;c)-f(1/\zeta;c^*)} \tag{19}$$

where c is any constant occurring in $f(\zeta)$ [12].

The idea of using the images of the singularities with respect to a circle in order to operate on the other contour without distorting the circle can surely be extended to problems dealing with two separate contours but not necessarily airfoils.

Two questions can be posed. First, how can Ives' technique produce the mapping, without using elliptic functions? The explanation can be found in the fact that the sequence of mappings used by Ferrari and Garrick consists of a logarithm, followed by an elliptic function. The net result actually has only one period, the other being neutralized by the multi-valuedness of the logarithm. On the other hand, the periodicity is introduced in Ives' mapping through the play of reflections of singularities produced by (19). The second question is whether one could bypass the two Theodorsen corrections, in the spirit of Section 3 above. Of course, in this case we would have to deal not with one distorted circle alone, but two, the first being particularly important since the images of (19) would, in any case, be defined with respect to a circle which now would only be an approximation to the real contour. I believe that the technique could still be applied, but no examples are available.

9. Cascades

The climax of difficulties is reached in the problem of generating a grid to compute flows through a cascade of airfoils. Garrick [13] sensed the difficulty well ahead of the computer era, but he did not have a way of measuring it; as we said, hand computations were necessarily limited to the simplest cases. He said: "It is to be noticed that improvements in the initial transformation are desirable and should be sought, particularly to take care of highly cambered airfoils more conveniently, in order to reduce the amount of subsequent calculations." At that time, the initial transformation was the one defined by (5), obviously a poor way to get started when dealing when highly cambered airfoils, since the basic shape furnished by (5) is a cascade of flat, straight, double-sided segments. Unfortunately, in a cascade four parameters must be considered, solidity, stagger, camber and thickness, and these parameters interact with each other, whatever the choice of the basic mapping. So long as solidity and stagger are low, even (5), applied to a moderately thick and cambered airfoil, produces a reasonable quasi-circle. But as solidity and stagger increase, the contour tends to become peanut-shaped (Fig. 15), with a catastrophic distribution of points around it (something like a wide circle, corresponding to a small portion of the original profile, with a small appendix, which is the image of all the rest); and, at times, the contour is not even star-shaped any longer. Can (5) be blamed for such a behavior? A detailed discussion of this question would transcend the limits of the present paper; let it just be said that the difficulty does not disappear when another mapping function, adopted by Legendre [33] and Ives [34], is used (Ives' function is the same as Legendre, to within a rotation and a bilinear transformation). This mapping can be written in the form:

$$\left[\frac{\zeta + b}{\zeta - b}\right]^{\delta} = \frac{\sin A(z - 1)}{\sin A(z + 1)} \qquad (20)$$

where z and ζ are the physical plane and the mapped plane, respectively, 1 and -1 are the location of the trailing edge and the center of the leading edge of a profile, respectively, δ is the usual Kármán-Trefftz exponent, A is related to solidity and stagger by

$$2\,i\,A = \pi\,s\,e^{i\beta} \qquad (21)$$

and

$$b = -\,i\,\tan\frac{A}{\delta} \qquad (22)$$

So far, the major advantages of this transformation with respect to (5) are the presence of an exponent, δ to produce profiles with a finite trailing edge angle, and the possibility of computing either ζ as a function of z, directly from (20), or z as a function of ζ from the inverse of (20):

$$z = -1 - \frac{i}{2A} \log \frac{e^{2iA} - w^{\delta}}{e^{-2iA} - w^{\delta}} \tag{23}$$

with

$$w = \frac{\zeta + b}{\zeta - b} \tag{24}$$

When stagger and solidity are high, most of the skill which must be used to follow the contour properly in applying (20) is wasted, due to the ungainly shape of the resulting contour. More manipulations are needed before a quasi-circular shape is obtained; but we will not elaborate on this point in the present paper.

Let us assume then, that a circle has been obtained somehow from the given cascade; and, in the spirit of Section 3, we will gladly accept a quasi-circle, without refining the mapping any further. At this stage, we must decide what type of grid we want. Different choices can indeed be made for the general appearance of the grid. Three of these are the most common (Fig. 16). The first contains lines which run from left to right through the cascade, more or less as streamlines of a real flow would do. The second (commonly called an O-grid) contains lines wrapped around each profile. The third (commonly called a C-grid) contains similar lines which run around the profile and an infinite line issuing from the trailing edge, as a wake. Now, regardless of the choice of the first mapping step, the plane of the circle will always have four singular points, representing the points at infinity (on the left and the right of the cascade) and their reflections on the circle. If we invert the circle plane so that the interior of the circle corresponds to the exterior of the cascade, then the two singular points inside the circle represent the physical points at infinity. A system of streamlines and equipotential lines for an incompressible, potential flow, proceeding from one singular point to the other and having stagnation points at the points on the circumference which correspond to the leading and trailing edge, provides a grid of the first type. In building it up, one finds numerical difficulties. It is indeed hard to follow a streamline on a spiral vortex when the singular point is very close to the periphery of the circle (this is what happens with high solidity and high stagger).

Polar coordinates are very simple to use, but they generate a very inconvenient grid because of the presence of a singularity in the flow field between profiles, at the point corresponding to the center of the circle [35]. (See Fig. 17.)

For an O-grid, the best procedure has been suggested by Ives [35]. It consists of mapping the circle onto a rectangle, in such a way that two vertices correspond to the singular points (points at infinity in the physical plane) and the opposite side corresponds to the circumference (the contour of the profile). The basic function defines the correspondence shown in Fig. 18 (where there is a straight slit between the two singularities). In the same figure, lines corresponding to straight lines, parallel to the sides of the rectangle, are shown inside the circle. The function which performs the task is the simplest of the Jacobian elliptic functions:

$$z = \mathrm{sn}\,(\zeta,\, m) \tag{25}$$

where m is defined by the position of the singularities and tends to 1 when solidity and stagger increase. Using Landen's transformations, the sine-amplitude can be expressed in terms of trigonometric functions (for m close to 0) or the hyperbolic tangent (for m close to 1); the coding of the subroutine is obvious and the computational time is negligible. Therefore, (25) is a convenient function.

Note that the circle (or quasi-circle) obtained through (5) or (20), and the circle obtained from (25) are by no means the same, despite the fact that in both planes the images of the points at infinity are symmetrically located on the real axis. The second circle is centered at the origin, but the first is not; therefore, a bilinear transformation must be used to map the two circles onto each other. This mapping depends on all four basic parameters and the trailing edge angle as well. Its interpretation, thus, is not easy, but a systematic study is needed to understand what range of basic contours, acceptable as a background of a grid for given profiles, can be obtained by using (25), a bilinear mapping and (23), in that sequence, without resorting to a Theodorsen step. It seems to me that we either have reasonable shapes, in which case that step can be skipped, or combinations of solidity and stagger which tend to generate intermediate shapes similar to the one on the right hand side of Fig. 15, in which case the Theodorsen technique would not work, and where another intermediate step, of a different nature, should be used. Even Joukowski mappings may help (see the last contour in Fig. 1). Work along these lines is urgently needed, if we want to obtain simple mapping procedures for three-dimensional turbomachinery problems. To make the point, we present a three-step sequence, in which the first circle, obtained from the rectangle, is the one of Fig. 18. A second circle (Fig. 19) is obtained by a bilinear transformation and the cascade, obtained through (23) is shown in Fig. 20. The contour is ugly, but the grid is perfectly usable.

10. Grids for ablated, three-dimensional bodies

We conclude this presentation showing a method applied to generate a computational grid for a three-dimensional, time-dependent problem. The flow to be determined is the shock layer around the ablated nose of a cone-cylinder; the flow is mostly supersonic, but it may have a subsonic bubble and an imbedded shock. The geometry of the body, which is axisymmetrical before ablation, becomes three-dimensional because of different ablation in different meridional planes. A grid is needed in a number of these planes, and corresponding points must be connected between adjacent planes, in order to generate a three-dimensional computational mesh. It is therefore convenient to have a grid defined in closed form, to make the evaluation of circumferential derivatives as easy as possible. For a given section of the body (Fig. 21) we define a skeleton, that is, a polygon, all contained inside the body, approximating the shape of the wall. Instead of using a Schwarz-Christoffel function to transform the skeleton into a straight line in a single operation, we have opted for straightening one vertex of the polygon at a time, beginning with the one farther from the nosetip. As in the case of the repeated Kármán-Trefftz mappings, the procedure is easily coded in a loop. The last step consists of a square root transformation, in order to bring the skeleton on the real axis and the body axis on the imaginary axis of the ζ-plane. The image of the wall on the ζ-plane is close to a straight line. The grid, which will be normalized between the image of the wall and the image of the bow shock, is always well shaped, despite strong concavities of the body produced by severe ablation. The method, first tested on axisymmetric problems [37], has been successfully applied to three-dimensional problems [38].

11. References

1. B. Riemann, Grundlagen für eine allgemeine Theorie der Functionen einer veränderlichen complexen Grösse, in "Collected works of Bernhard Riemann", Dover Publ., New York 1953.

2. H.A. Schwarz, Gesammelte Abhandlungen, Berlin 1890.

3. G.R. Kirchhoff, Zur Theorie freier Flüssigkeitsstrahlen, in "Gesammelte Abhandlungen", Leipzig 1881.

4. H. Lamb, Hydrodynamics, Dover Publ., New York 1945.

5. N. Joukowski, Ueber die Konturen der Drachenflieger, ZFM 1910.

6. T. von Kármán and E. Trefftz, Potentialströmungen um gegebene Tragflächen-Querschnitte, ZFM 1918.

7. E. König, Potentialströmung durch Gitter, ZAMM 2, 422, 1922.

8. C. Ferrari, Sulla trasformazione conforme di due cerchi in due profili alari, Mem. Acc. Sci. Torino (2), 67, 1933.

9. T. Theodorsen, Theory of wing sections of arbitrary shape, N.A.C.A. TR 411, 1931.

10. S.E. Warschawski, On Theodorsen's method of conformal mapping on nearly circular regions, Quart. Appl. Math., 3, 12, 1945.

11. T. Theodorsen and I.E. Garrick, General potential theory of arbitrary wing sections, N.A.C.A. TR 452, 1933.

12. I.E. Garrick, Potential flow about arbitrary biplane wing sections, N.A.C.A. TR 542, 1936.

13. I.E. Garrick, Conformal mapping in Aerodynamics, with emphasis on the method of successive conjugates, Symp. on construct. and appl. of conf. maps., Nat. Bureau of Standards, Appl. Math. Series 18, 137, 1949.

14. H.B. Phillips and N. Wiener, Nets and the Dirichlet problem, J.Math.Phys. 2, 105, 1923.

15. J.F. Thompson, F.C. Thames, C.W. Martin and S.P. Shanks, Use of numerically generated body-fitted coordinate systems for solution of the Navier-Stokes equations, Proc. AIAA 2nd Comp. Fl. Dyn. Conf., 10, 1975.

16. G. Moretti, B. Grossman and F. Marconi, Jr., A complete numerical technique for the calculation of three-dimensional inviscid supersonic flows, AIAA Paper 72-192, 1972.

17. G. Moretti, Conformal mappings for computations of steady, three-dimensional, supersonic flows, Numerical/Laboratory Comp. Methods in Fl. Mech., ASME, 13, 1976.

18. A. Jameson, Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1, Comm. Pure Appl. Math. 27, 283,1974.

19. D.A. Caughey, A systematic procedure for generating useful conformal mappings, Int. J. Num. Meth. in Eng., 12, 1651, 1978.

20. G.T. Symm, An integral equation method in conformal mapping, Numer. Math. 9, 250, 1966.

21. J.K. Hayes, D.K. Kahaner and R.G. Kellner, An improved method for numerical conformal mapping, Math. of Comp. 26, 327, 1972.

22. J.L. van Ingen, Advanced computer technology in aerodynamics, AGARD LS 16, 1969.

23. R.T. Davis, Numerical methods for coordinate generation based on Schwarz-Christoffel transformations, AIAA Comp.F.Dyn.Conf., Williamsburg, Va. 1979, page 180.

24. G. Birkhoff, D.M. Young and E.H. Zarantonello, Numerical methods in conformal mapping, Proc. Symp. in Appl. Math., vol. 4, Fl. Dyn, McGraw-Hill, Publ. 1953, page 117.

25. R.S. Skulsky, A conformal mapping method to predict low-speed aerodynamic characteristics of arbitrary slender reentry shapes, J. Spacecraft 3, 247, 1966.

26. L.N. Trefethen, Numerical computation of the Schwarz-Christoffel transformation, Stanford U. Rep. STAN-CS-79-710, 1979.

27. V.J. Rossow, Conformal mapping for potential flow about airfoils with attached flap, J. Aircraft 10, 60, 1973.

28. G. Moretti, A numerical analysis of muzzle blast precursor flow, POLY M/AE Rep. 80-10, 1980.

29. H. Kober, Dictionary of conformal representations, Dover Publ., New York, 1952.

30. D.A. Caughey and A. Jameson, Accelerated iterative calculation of transonic nacelle flowfields, AIAA J. 15, 1474, 1977.

31. B.G. Arlinger, Calculation of transonic flow around axisymmetric inlets, AIAA J. 13, 1614, 1975.

32. D.C. Ives, A modern look at conformal mapping, including multiply connected regions, AIAA J. 14, 1006, 1976.

33. R.G. Legendre, Work in progress in France related to computation of profiles for turbomachine blades by hodograph method, ASME Paper 72-gt-41, 1972.

34. D.C. Ives and J.F. Liutermoza, Analysis of transonic cascade flow using conformal mapping and relaxation techniques, AIAA J. 15, 647, 1977.

35. D.A. Frith, Inviscid flow through a cascade of thick, cambered airfoils, ASME Papers 73-gt-84 and 73-gt-85, 1973.

36. G. Moretti, Computation of shock layers about ablated, blunt-nosed bodies, POLY Rep. 77-14, 1977.

37. D.W. Hall, Calculation of inviscid supersonic flow over ablated nosetips, AIAA Paper 79-0342, 1979.

This research was conducted, in part, under the sponsorship of the NASA Langley Research Center under Grant No. NSG 1248.

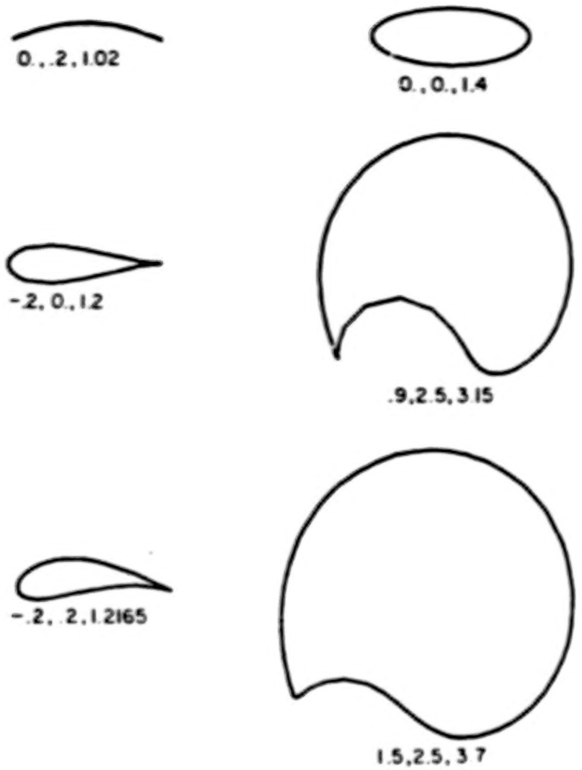

0.,.2,1.02
0.,0.,1.4
-.2,0.,1.2
.9,2.5,3.15
-.2,.2,1.2165
1.5,2.5,3.7

Figure 1.- Joukowski contours for different values of Real(c), Imag(c), r.

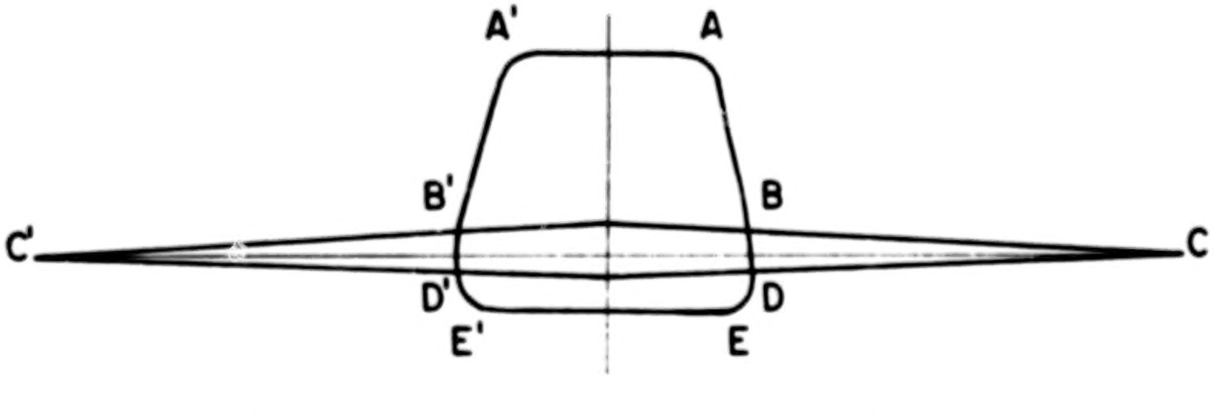

A'
A
B'
B
C'
C
D'
D
E'
E

Figure 2.- Cross-section of fuselage and wing.

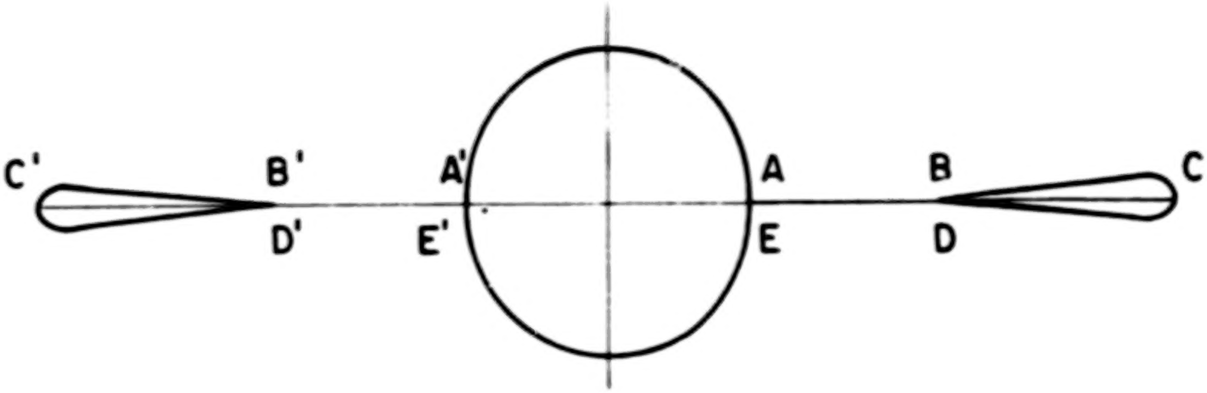


Figure 3.- Cross-section of fuselage and arrow wing.

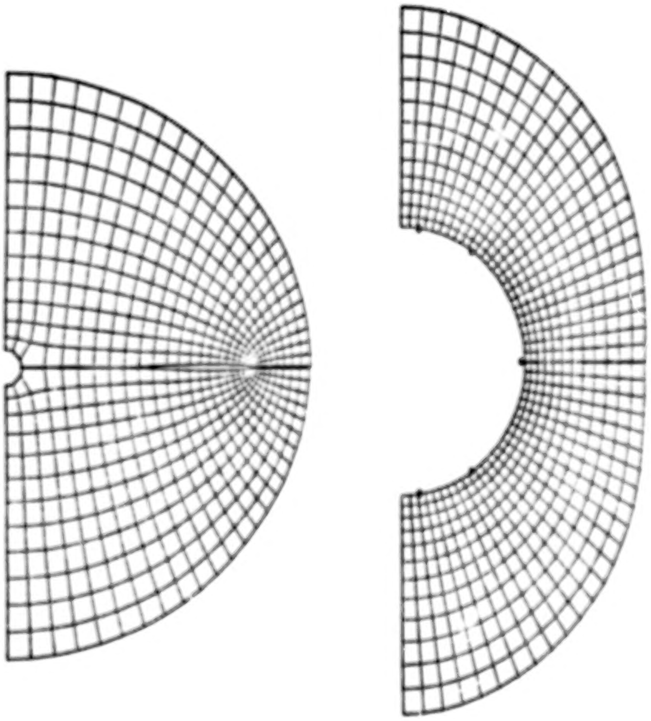


Figure 4.- Grid for fuselage-and-arrow-wing calculation, in physical plane and mapped plane.

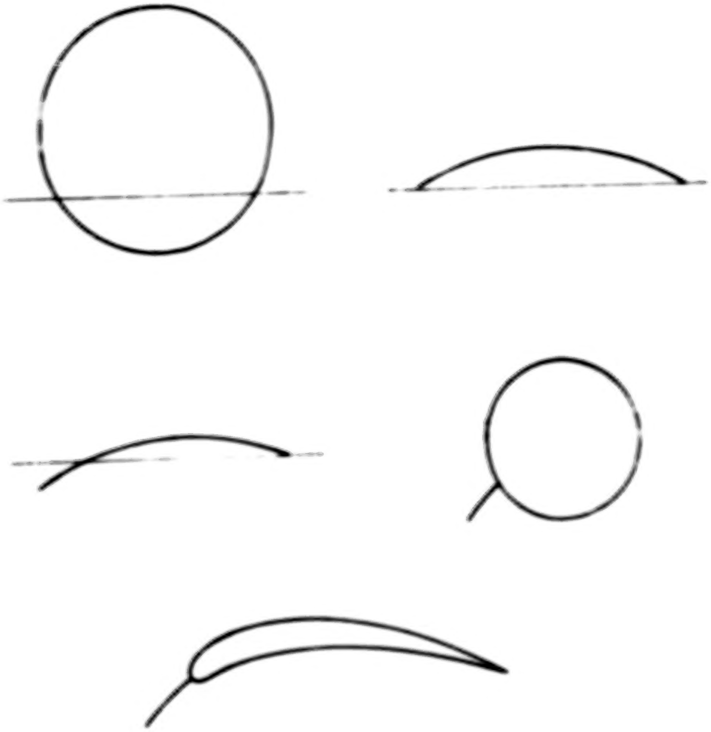


Figure 5.- Generation of wing with attached flap.

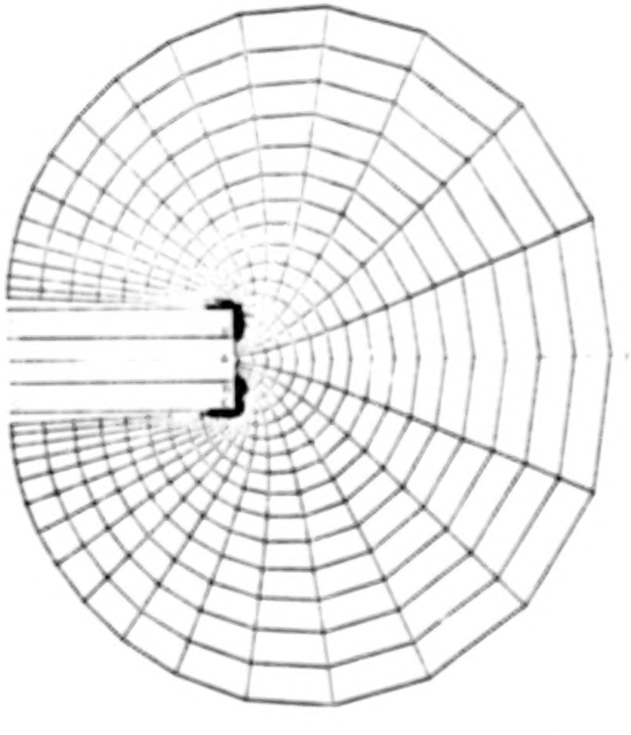


Figure 6.- Grid for muzzle blast calculation.

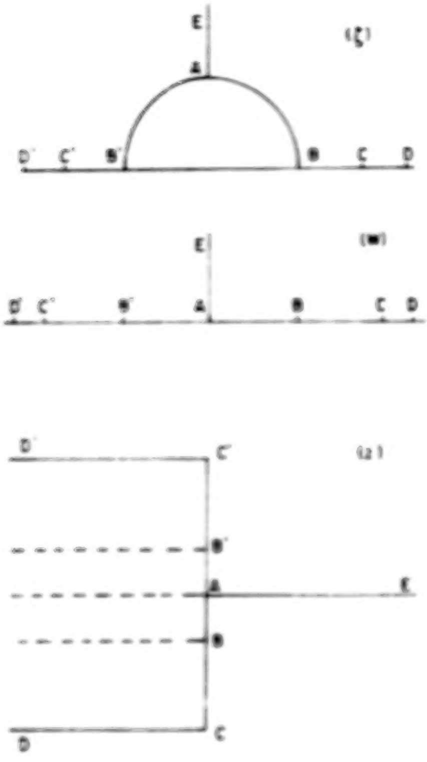

(ζ)
E
A
D'
C'
B'
B
C
D
(w)
E
D'
C'
B'
A
B
C
D
(z)
D'
C'
B'
A
E
B
C
D

Figure 7.- Generation of the muzzle mapping.

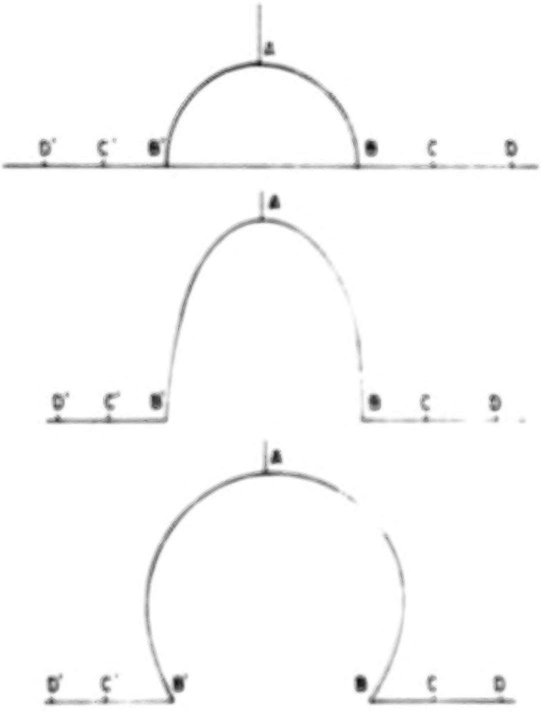

A
D'
C'
B'
B
C
D
A
D'
C'
B'
B
C
D
A
D'
C'
B'
B
C
D

Figure 8.- Generation of the muzzle mapping,
with protruding bullet.

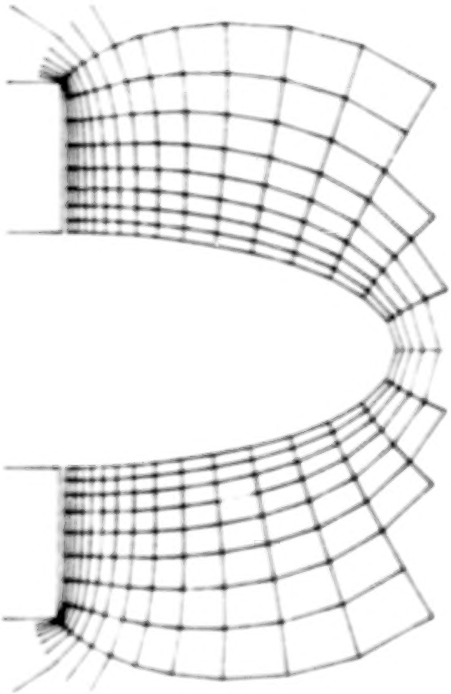


Figure 9.- Grid for muzzle with protruding bullet.

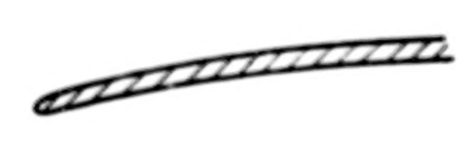


Figure 10.- Nacelle geometry.

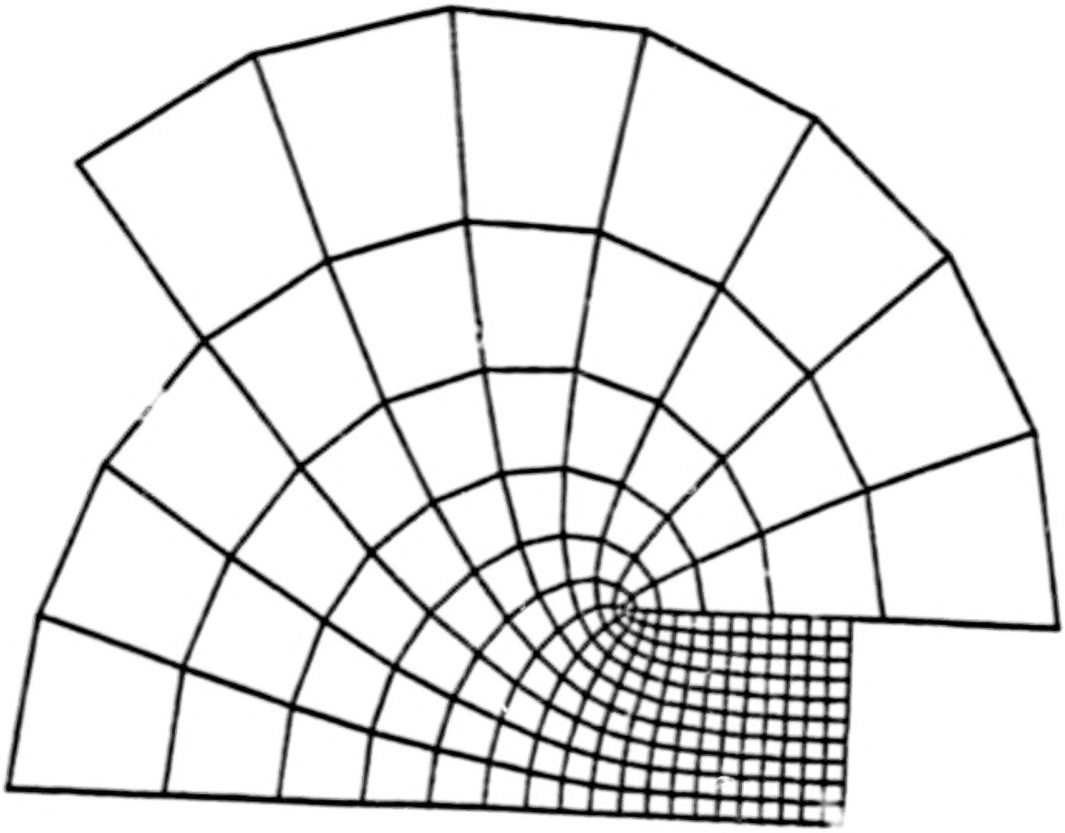


Figure 11.- Grid generated by equation (16).

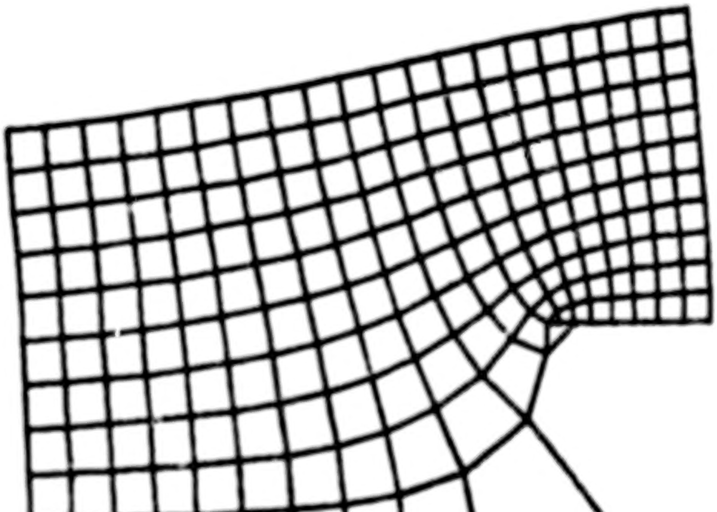


Figure 12.- Grid generated by equation (17), Cartesian coordinates.

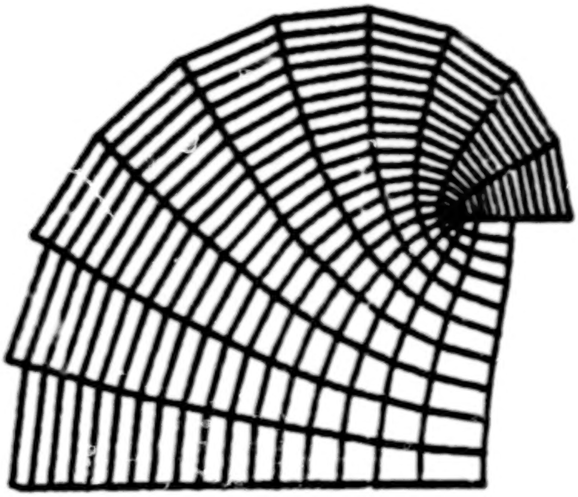


Figure 13.- Grid generated by equation (17), polar coordinates.

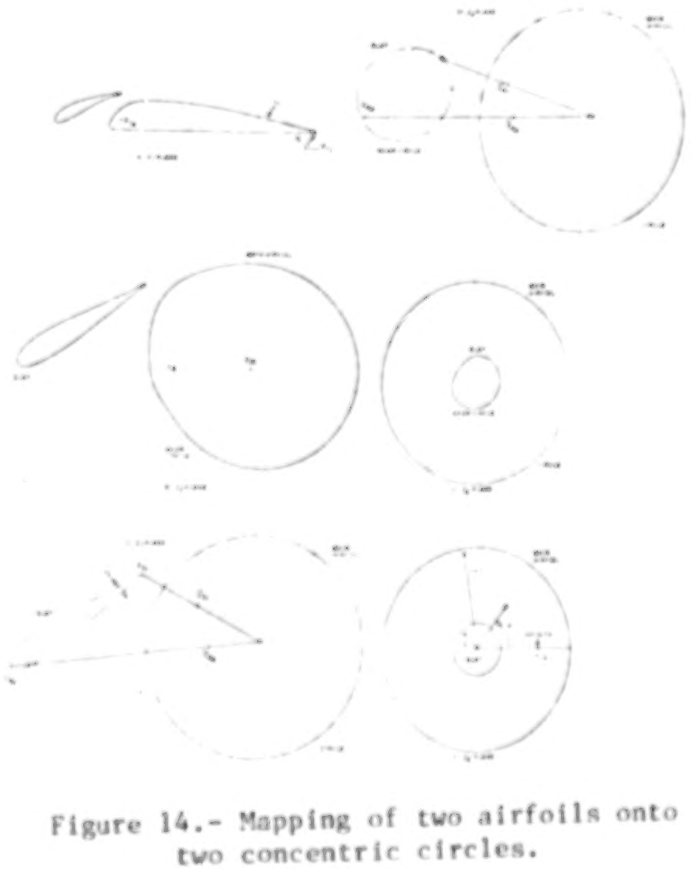


Figure 14.- Mapping of two airfoils onto two concentric circles.

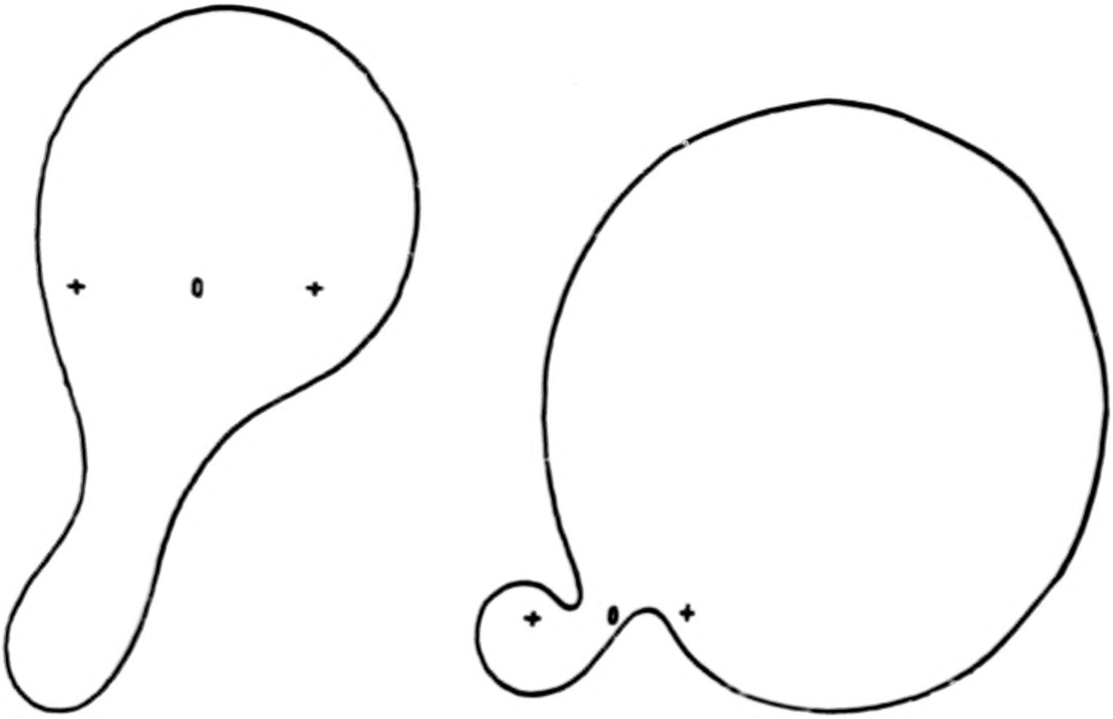


Figure 15.- "Quasi-circles" mapped from cascades of high solidity.

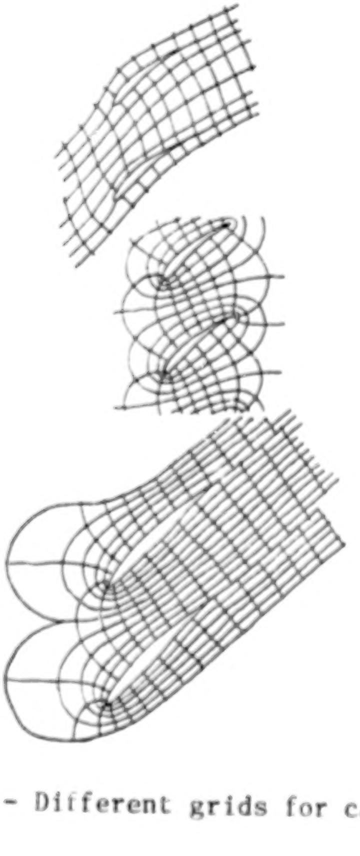


Figure 16.- Different grids for cascades.

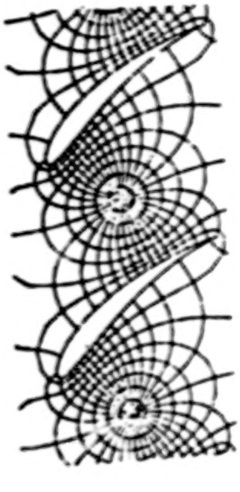


Figure 17.- Frith's grid for cascades.

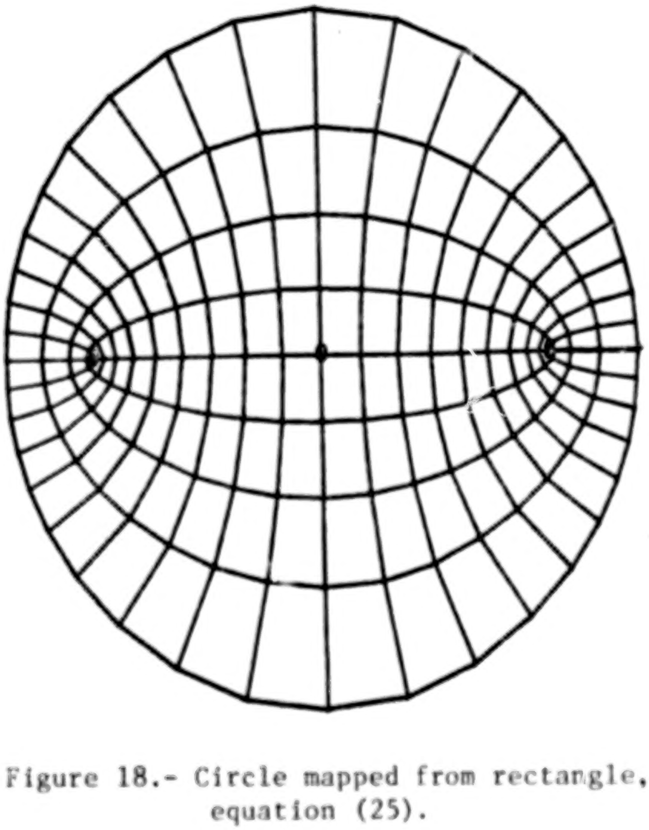


Figure 18.- Circle mapped from rectangle, equation (25).

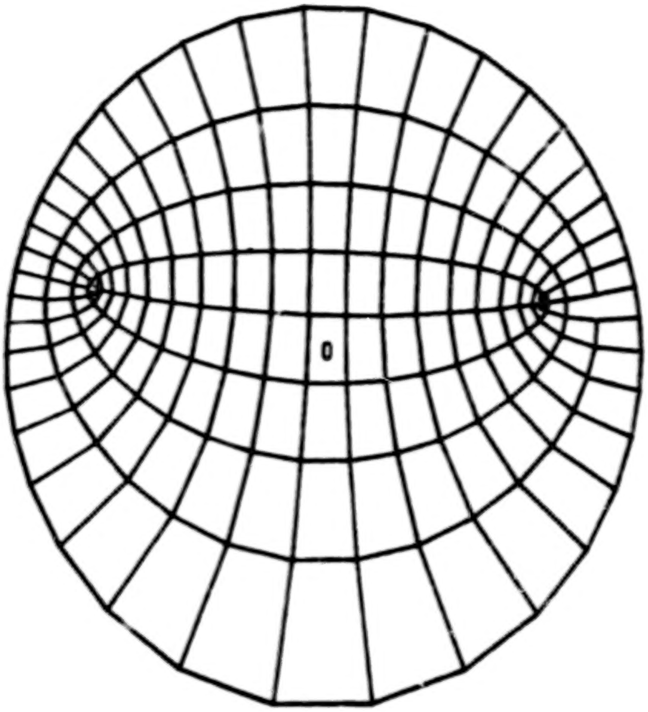


Figure 19.- Circle mapped from cascade, equation (20).

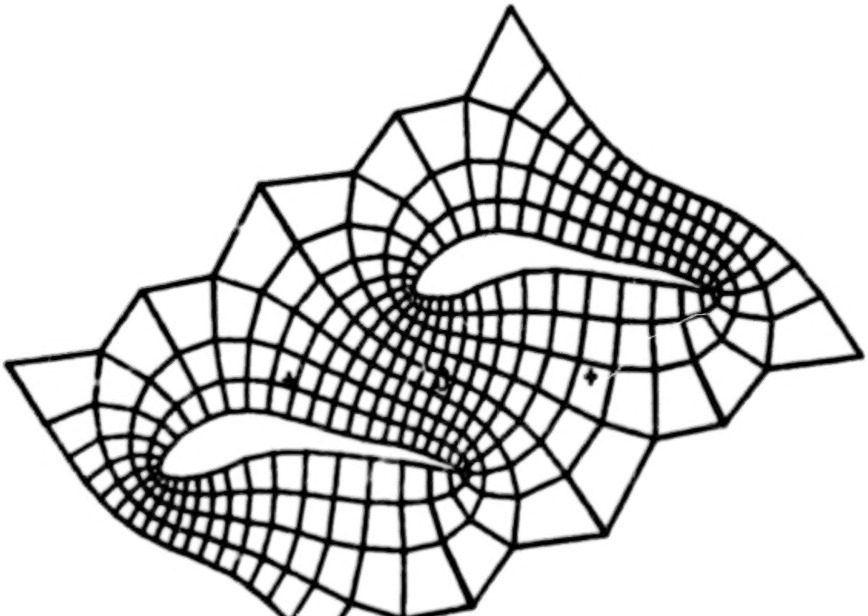


Figure 20.- Cascade obtained from circle of Figure 19.

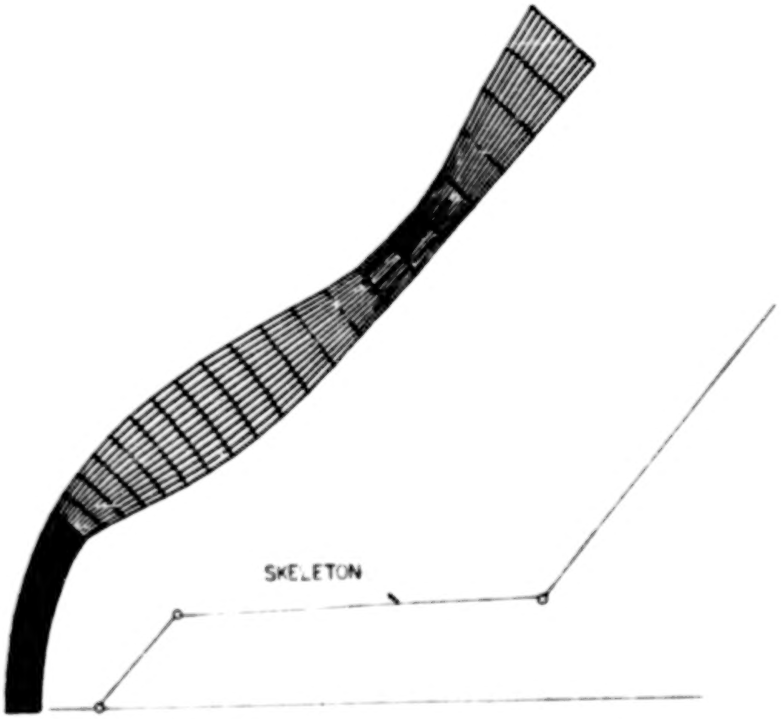


Figure 21.- Grid for ablated body.

BLANK PAGE

BLANK PAGE

GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES*

Joe F. Thompson and C. Wayne Mastin
Department of Aerospace Engineering
Department of Mathematics
Mississippi State University
Mississippi State, MS 39762

I. INTRODUCTION

In recent years a multitude of techniques has been developed for generating computational grids required in the finite difference or finite element solutions of partial differential equations on arbitrary regions. The importance of the choice of the grid is well known. A poorly chosen grid may cause results to be erroneous or may fail to reveal critical aspects of the true solution. Some considerations that are involved in grid selection can be noted from the papers of Blottner and Roache [1], Crowder and Dalton [2], and Kalnay de Rivas [3]. While these papers discuss error for one-dimensional problems, few results exist for higher dimensions. This report will examine the errors in approximating the derivatives of a function by traditional central differences at grid points of a curvilinear coordinate system. The implications concerning the accuracy of the numerical solution of a partial differential equation will be explained by considering several numerical examples. Although this study only considers the two-dimensional case, the techniques and implications are equally valid for three-dimensional grids.

An interesting feature of the error analysis in this report is its simplicity. Most of the results follow by merely working with the truncation terms of some power series expansion. It is noted that these series expansions also give rise to higher-order difference approximations which can significantly reduce error when the grid spacing changes rapidly, as might be the case in problems with shock waves or thin boundary layers.

---

*This research was sponsored by NASA Langley Research Center under Grants NSG 1577 and NGR-25-001-055.

When transforming a partial differential equation from rectangular to curvilinear coordinates, the derivatives of the functions defining the transformation must be evaluated. If the relation between rectangular and curvilinear variables is given by a simple analytic expression, the transformation derivatives may be computed either analytically or numerically. Truncation errors in both cases are considered for comparison.

One objective of this work is to provide tools to examine a grid, together with a computed solution, and predict possible inaccuracies due to the grid. The grid may thus be redefined to give a better solution. Directions for future work could be an extension to higher dimensions of the one-dimensional grid optimization technique of Pierson and Kutler [4].

This report also discusses the control of coordinate line spacing through functions incorporated in the elliptic generating system for the curvilinear coordinates. Attraction of coordinate lines to other coordinate lines and also attraction to fixed lines in physical space are covered. Appropriate forms of the control functions required to produce desired spacing distributions are derived. Finally a procedure for distribution of points around a boundary curve according to local boundary curvature is given. In addition a few examples of recent generation of coordinate systems are given.

II. TRUNCATION ERROR ANALYSIS

Suppose a curvilinear coordinate system is generated by transforming an arbitrary physical region of the xy-plane onto a rectangular computational region of the $\xi\eta$-plane. The relationship between partial derivatives of a function f with respect to physical and computational variables is well-known. It will be included here for later comparison with approximations derived from series expansions. Only first and second order derivatives will be considered:

$$\frac{\partial f}{\partial \xi} = \frac{\partial x}{\partial \xi}\frac{\partial f}{\partial x} + \frac{\partial y}{\partial \xi}\frac{\partial f}{\partial y}$$

$$\frac{\partial^2 f}{\partial \xi^2} = \frac{\partial^2 x}{\partial \xi^2}\frac{\partial f}{\partial x} + \frac{\partial^2 y}{\partial \xi^2}\frac{\partial f}{\partial y} + \left(\frac{\partial x}{\partial \xi}\right)^2 \frac{\partial^2 f}{\partial x^2} + 2\,\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi}\frac{\partial^2 f}{\partial x \partial y} + \left(\frac{\partial y}{\partial \xi}\right)^2 \frac{\partial^2 f}{\partial y^2} \qquad (1)$$

$$\frac{\partial^2 f}{\partial \xi \partial \eta} = \frac{\partial^2 x}{\partial \xi \partial \eta}\frac{\partial f}{\partial x} + \frac{\partial^2 y}{\partial \xi \partial \eta}\frac{\partial f}{\partial y} + \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial^2 f}{\partial x^2} + \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\frac{\partial^2 f}{\partial x \partial y} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial^2 f}{\partial y^2}$$

The derivatives with respect to η can be obtained by replacing ξ with η in the first two equations of (1).

Although this change of variables formulation can be easily used in deriving difference approximations for derivatives with respect to x and y, nothing can be said about truncation error. An error analysis can, however, be based on Taylor series expansion of function values at neighboring points about a single point in the physical region. In order to distinguish between derivatives and differences in the following, the differential notation is used for derivatives while subscripts denote the usual second order central difference expressions. The following approximations for the central differences are valid when all series are truncated after second derivative terms. A unit mesh width in the $\xi\eta$-plane is assumed without loss of generality.

$$f_\xi = x_\xi \frac{\partial f}{\partial x} + y_\xi \frac{\partial f}{\partial y} + \frac{1}{2} x_\xi x_{\xi\xi} \frac{\partial^2 f}{\partial x^2} + \frac{1}{2}(x_\xi y_{\xi\xi} + y_\xi x_{\xi\xi}) \frac{\partial^2 f}{\partial x \partial y}$$

$$+ \frac{1}{2} y_\xi y_{\xi\xi} \frac{\partial^2 f}{\partial y^2}$$

$$f_{\xi\xi} = x_{\xi\xi} \frac{\partial f}{\partial x} + y_{\xi\xi} \frac{\partial f}{\partial y} + (x_\xi{}^2 + \frac{1}{4} x_{\xi\xi}{}^2) \frac{\partial^2 f}{\partial x^2} + 2(x_\xi y_\xi + \frac{1}{4} x_{\xi\xi} y_{\xi\xi})$$

$$+ \frac{\partial^2 f}{\partial x \partial y} + (y_\xi{}^2 + \frac{1}{4} y_{\xi\xi}{}^2) \frac{\partial^2 f}{\partial y^2} \qquad (2)$$

$$f_{\xi\eta} = x_{\xi\eta} \frac{\partial f}{\partial x} + y_{\xi\eta} \frac{\partial f}{\partial y} + \frac{1}{2}((x^2)_{\xi\eta} - 2xx_{\xi\eta}) \frac{\partial^2 f}{\partial x^2}$$

$$+ ((xy)_{\xi\eta} - xy_{\xi\eta} - yx_{\xi\eta}) \frac{\partial^2 f}{\partial x \partial y} + \frac{1}{2}((y^2)_{\xi\eta} - 2yy_{\xi\eta}) \frac{\partial^2 f}{\partial y^2}$$

Together with the corresponding two equations for f_η and $f_{\eta\eta}$, this constitutes a system of five simultaneous equations which can be solved to produce difference expressions of two first and three second derivatives of f with respect to x and y. Assuming the third order derivatives of f are bounded, the truncation error in the above expressions is $O(h^3)$, where h is some measure of the local mesh spacing. Consequently, when (2) is solved for the difference approximations of the physical derivatives of f, the truncation error is $O(h^2)$ for first derivatives and $O(h)$ for second order derivatives. In contrast, solving the system (1) with the f, x, and y derivatives replaced by differences (and including the corresponding equations for f_η and $f_{\eta\eta}$) simultaneously to produce expressions for the five physical derivatives of f gives rise to $O(h)$ and $O(1)$ truncation errors for the first and second order derivatives.

In both cases it has been assumed that the coefficient matrices on the right hand sides of (1) and (2), i.e., the coordinate derivatives, including the omitted η differences, are well conditioned. Ill conditioned matrices which may result from extremely skewed coordinate lines could cause further deterioration in accuracy. Higher order accuracy can be obtained using (1) if second order coordinate differences are assumed to be $O(h^2)$. This effectively limits the rate of change in coordinate line spacing and the curvature of coordinate lines, however. No simple relation between the coefficients of the second order derivatives in the last equation of (1) and (2) was found except for the fact that they would be equal if the differences in (2) were replaced by derivatives.

The variation in numerical solutions using (1) and (2) is illustrated in the solution of Laplace's equation. The function

$$u(x, y) = x(1 + 1/(x^2 + y^2))$$

satisfies Laplace's equation for $x^2 + y^2 > 1$ and has a vanishing normal derivative on the boundary. This boundary value problem was solved numerically on $1 \leq x^2 + y^2 \leq 100$. A grid was selected with 39 radial coordinate lines and 49 circular coordinate lines. The first 23 circular coordinate lines were uniformly spaced after which the spacing was increased by a factor of 5. The difference between the exact and numerical solution is indicated in Figure 1 for difference equations derived from (1) and (2). The effect of the sudden change in coordinate line spacing was clearly less severe when using difference expressions from the higher order series expansion.

A similar error analysis can be carried out where the derivatives of x and y with respect to ξ and η are computed analytically rather than approximated by differences. In this case a series expansion in the $\xi\eta$-plane is required, followed by substitution of expressions for the higher-order ξ and η derivatives in terms of the x and y derivatives (see Ref. 5 for complete detail). Retaining physical derivatives of f through second order, as in (2), the following approximations are generated. The second derivative approximations, $f_{\xi\xi}$ and $f_{\xi\eta}$, are very lengthy and only the first and second order derivatives of x and y are included here, the complete expressions being given in Ref. 5. The first derivative approximation includes third order derivatives:

$$
\begin{aligned}
f_\xi &= \left(\frac{\partial x}{\partial \xi} + \frac{1}{6}\frac{\partial^3 x}{\partial \xi^3}\right)\frac{\partial f}{\partial x} + \left(\frac{\partial y}{\partial \xi} + \frac{1}{6}\frac{\partial^3 y}{\partial \xi^3}\right)\frac{\partial f}{\partial y} + \frac{1}{2}\frac{\partial x}{\partial \xi}\frac{\partial^2 x}{\partial \xi^2}\frac{\partial^2 f}{\partial x^2} \\
&\quad + \frac{1}{2}\left(\frac{\partial x}{\partial \xi}\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial y}{\partial \xi}\frac{\partial^2 x}{\partial \xi^2}\right)\frac{\partial^2 f}{\partial x \partial y} + \frac{1}{2}\frac{\partial y}{\partial \xi}\frac{\partial^2 y}{\partial \xi^2}\frac{\partial^2 f}{\partial y^2} \\
f_{\xi\xi} &= \frac{\partial^2 x}{\partial \xi^2}\frac{\partial f}{\partial x} + \frac{\partial^2 y}{\partial \xi^2}\frac{\partial f}{\partial y} + \left(\left(\frac{\partial x}{\partial \xi}\right)^2 + \frac{1}{4}\left(\frac{\partial^2 x}{\partial \xi^2}\right)^2\right)\frac{\partial^2 f}{\partial x^2} \\
&\quad + 2\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} + \frac{1}{4}\frac{\partial^2 x}{\partial \xi^2}\frac{\partial^2 y}{\partial \xi^2}\right)\frac{\partial^2 f}{\partial x \partial y} + \left(\left(\frac{\partial y}{\partial \xi}\right)^2 + \frac{1}{4}\left(\frac{\partial^2 y}{\partial \xi^2}\right)^2\right)\frac{\partial^2 f}{\partial y^2} \qquad (3) \\
f_{\xi\eta} &= \frac{\partial^2 x}{\partial \xi \partial \eta}\frac{\partial f}{\partial x} + \frac{\partial^2 y}{\partial \xi \partial \eta}\frac{\partial f}{\partial y} + \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{1}{2}\frac{\partial^2 x}{\partial \xi \partial \eta}\left(\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2}\right)\right)\frac{\partial^2 f}{\partial x^2} \\
&\quad + \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} - \frac{1}{2}\frac{\partial^2 x}{\partial \xi \partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2}\right) - \frac{1}{2}\frac{\partial^2 y}{\partial \xi \partial \eta}\right. \\
&\quad \left.\left(\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2}\right)\right)\frac{\partial^2 f}{\partial x \partial y} + \left(\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{1}{2}\frac{\partial^2 y}{\partial \xi \partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2}\right)\right)\frac{\partial^2 f}{\partial y^2}
\end{aligned}
$$

Considerable similarity exists between the approximations in (2) and (3) and corresponding statements can be made about the effects of the coordinate system on truncation error. For example, it can be noted that for the first derivative approximations to be second order accurate, the second and third order derivatives of x and y must be $O(h^2)$ and $O(h^3)$, respectively. Due to the additional restriction on the third order derivatives, it is not difficult to find examples where solutions of (1) with numerically computed derivatives of x and y are much more accurate than solutions using the analytical expressions for these derivatives.

With reasonable care in the selection of the grid any of the above difference formulations will give equally good results. For example, consider the grid for the region about a Joukowski airfoil depicted in Figure 2. This grid was constructed by the conformal mapping of an annular region with uniformly spaced circular coordinates. As in the above example, Laplace's equation is solved with vanishing normal derivative imposed on the airfoil. The solution is the velocity potential for flow about the airfoil at zero angle of attack. Table 1 indicates the

the difference between the computed solution and the exact solution on the surface of the airfoil where the error was greatest.

Table 1. Comparison of Difference Formulations

Differencing Method	Max Error	RMS Error
Taylor Series (2)	.03123	.00864
Analytic (1)	.02216	.01256
Numerical (1)	.02411	.00795

For this example there is clearly no advantage in using the difference expression from the series expansion in (2) over using (1) with the derivatives of x and y computed either analytically or numerically. There is another aspect to the question of the use of analytically calculated coordinate derivatives, as opposed to numerical difference representatives, when fully conservative difference formulations are used. In that case the formulation will not be fully conservative with the analytical expression in the sense that a uniform solution on the field will not be strictly preserved. This can lead to instability if the differences of the coordinate derivatives are large.

Thus far only problems of error which deal directly with the coordinate system have been considered. This source of error can be controlled by limiting the higher order differences of derivatives of x and y. A more serious problem in numerical computations is the error in the approximate solution which results from large higher order derivatives of f. In transforming from physical to computational variables, the derivatives of f with respect to ξ and η are replaced by differences regardless of whether derivatives or differences are used for x and y. The truncation error in approximating the computational derivatives of f can be minimized to some degree by a properly chosen grid. However, there are limitations in the grid choice since, as we have previously observed, a highly distorted grid also contributes to large truncation errors in the approximation of the physical derivatives of f. To analyze the total truncation error due to solution and grid, it is convenient to introduce matrix notation.

Suppose the derivatives in the physical and computational planes are related by (1). This relation can be written

$$\Delta = AD \qquad (4)$$

where

$$\Delta = \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial^2 f}{\partial \xi^2} \\ \frac{\partial^2 f}{\partial \xi \partial \eta} \\ \frac{\partial^2 f}{\partial \eta^2} \end{bmatrix}, \quad D = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad A = \left[\begin{array}{c|c|c|c|c} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & 0 & 0 & 0 \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & 0 & 0 & 0 \\ \frac{\partial^2 x}{\partial \xi^2} & \frac{\partial^2 y}{\partial \xi^2} & (\frac{\partial x}{\partial \xi})^2 & 2\,\frac{\partial x}{\partial \xi}\,\frac{\partial y}{\partial \xi} & (\frac{\partial y}{\partial \xi})^2 \\ \frac{\partial^2 x}{\partial \xi \partial \eta} & \frac{\partial^2 y}{\partial \xi \partial \eta} & \frac{\partial x}{\partial \xi}\,\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi}\,\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\,\frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \xi}\,\frac{\partial y}{\partial \eta} \\ \frac{\partial^2 x}{\partial \eta^2} & \frac{\partial^2 y}{\partial \eta^2} & (\frac{\partial x}{\partial \eta})^2 & 2\,\frac{\partial x}{\partial \eta}\,\frac{\partial y}{\partial \eta} & (\frac{\partial y}{\partial \eta})^2 \end{array}\right]$$

Difference expressions for D are generated by replacing the elements of Δ (and possibly A) by the appropriate difference approximations. If the truncation term is retained, the equation (4) becomes

$$\delta + \epsilon = AD \tag{5}$$

where δ is the vector containing the difference approximations and

$$\epsilon = \begin{bmatrix} -\frac{1}{6}\,\frac{\partial^3 f}{\partial \xi^3} \\ -\frac{1}{6}\,\frac{\partial^3 f}{\partial \eta^3} \\ -\frac{1}{12}\,\frac{\partial^4 f}{\partial \xi^4} \\ \frac{}{6}(\frac{\partial^4 f}{\partial \xi^3 \partial \eta} + \frac{\partial^4 f}{\partial \xi \partial \eta^3}) \\ -\frac{1}{12}\,\frac{\partial^4 f}{\partial \eta^4} \end{bmatrix}$$

Solving for D in (5) we have

$$D = A^{-1}\delta + A^{-1}\varepsilon \qquad (6)$$

Now ε is unknown but can be estimated using differences of f. Although such numerical differentiation does not tend to be very accurate when applied to an approximate solution of a partial differential equation, the value of $A^{-1}\varepsilon$ has been used successfully to distinguish regions of high error from regions of low error. This can be illustrated by returning to the numerical solution of potential flow about the Joukowski airfoil in Figure 2. The comparison of truncation error with error in the solution is indicated in Figure 3 for grid points beginning near the trailing edge and ending near the leading edge of the airfoil. The grid points were chosen to lie on the second coordinate line from the airfoil surface so that no extrapolation was needed to estimate the elements of ε.

Each factor in the truncation error estimate can be analyzed independently. The factor A^{-1} deals only with the grid coordinates, while ε involves only the solution of the partial differential equation. In the above example consideration of ε alone would seriously underestimate the order of accuracy near the leading and trailing edge since the distortion in the coordinate system would not be taken into account. The influence of the factor A^{-1} can be analyzed by examining the condition of the matrix A. An ill-conditioned matrix not only magnifies the effect of the truncation terms in ε but also the effect of deleting the additional terms which appeared in the series expansions (2) and (3).

We will now consider a case where an extremely ill-conditioned matrix is encountered. The Navier-Stokes equations in stream function-vorticity formulation were solved numerically for viscous flow about a circular cylinder. The data in Table 2 illustrates the growth in the condition number of A as the circular coordinate lines are concentrated near the cylinder to resolve the boundary layer. Only the Laplacian of vorticity was included in the truncation error computation since this truncation term clearly dominated the remaining truncation terms in the equations. As n increases, the dominating factors in the truncation term shifts from the elements of ε to the elements of A^{-1}. An examination of vorticity values revealed a clear deterioration of the numerical solution for $n = 4$.

Table 2. Maximum truncation error for $\nabla^2\omega$ and condition number of A. Circular coordinate lines $r = 1 + 9(1 - \exp(n\eta/48)) / (1 - \exp(n))$, $\eta = 0, 1, ---, 48$. Reynolds no. = 5.

n	Max Truncation	Max Cond_1 (A)
1	.1079	45
2	.0482	78
3	.0891	259
4	3.0918	1017

For later reference, we have from (2) for the one-dimensional case that the simple two-point central difference expression for the first derivative, f_ξ/x_ξ, has a truncation error term given by

$$\frac{1}{2}x_{\xi\xi}\frac{\partial^2 f}{\partial x^2}$$

which acts as a numerical diffusion. This effect was pointed out earlier in Ref. 1.

III. COORDINATE SYSTEM CONTROL

A. Original Generating System

In the formulation of boundary-fitted coordinate systems generated from elliptic systems as given in Ref. 6 the curvilinear coordinates (ξ, η) were determined as the solution of the system

$$\nabla^2 \xi = P(\xi,\eta) \tag{7a}$$

$$\nabla^2 \eta = Q(\xi,\eta) \tag{7b}$$

which in the transformed plane becomes (from here on, subscripts indicate derivatives)

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_{\xi} + Qx_{\eta}) \tag{8a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Py_{\xi} + Qy_{\eta}) \tag{8b}$$

with

$$\alpha = x_{\eta}^{2} + y_{\eta}^{2} \tag{9a}$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta} \tag{9b}$$

$$\gamma \equiv x_\xi^{\,2} + y_\xi^{\,2} \tag{9c}$$

$$J \equiv x_\xi y_\eta - x_\eta y_\xi \tag{9d}$$

B. Attraction to Coordinate Lines

Here the functions P and Q are to be chosen to control the coordinate line spacing. In Ref. 6 those control functions were taken as sums of decaying exponentials of the form

$$P = -\sum_{i=1}^{n} a_i \operatorname{sgn}(\xi - \xi_i)\exp(-c_i|\xi - \xi_i|) \tag{10a}$$

$$- \sum_{i=1}^{m} b_i \operatorname{sgn}(\xi - \xi_i)\exp(-d_i((\xi - \xi_i)^2 + (\eta - \eta_i)^2)^{1/2})$$

$$Q = -\sum_{i=1}^{n} a_i \operatorname{sgn}(\eta - \eta_i)\exp(-c_i|\eta - \eta_i|) \tag{10b}$$

$$- \sum_{i=1}^{m} b_i \operatorname{sgn}(\eta - \eta_i)\exp(-d_i((\xi - \xi_i)^2 + (\eta - \eta_i)^2)^{1/2})$$

Here the a_i, b_i, c_i, and d_i of the Q functions are not necessarily the same as those in the P function.

In the P function the effect of the amplitude a_i is to attract ξ-

coordinate lines toward the ξ_i-line, while the effect of the amplitude b_i is to attract ξ-lines toward the single point (ξ_i, η_i). Note that this attraction to a point is actually attraction of ξ-lines to a point on another ξ-line, and as such acts normal to the ξ-line through the point. There is no attraction of η-lines to this point via the P function. In each case the range of the attraction effect is determined by the decay factors, c_i and d_i. With the inclusion of the sign changing function, the attraction occurs on both sides of the ξ-line, or the (ξ_i, η_i) point, as the case may be. Without this function, attraction occurs only on the side toward increasing ξ, with repulsion occuring on the other side.

A negative amplitude simply reverses all of the above-described effects, i.e., attraction becomes repulsion and vice versa. The effect of the Q function on η-lines follows analagously. A number of examples of this type of coordinate line control have been given in Ref. 6.

In the case of a boundary that is an η-line, positive amplitudes in the Q function will cause η-lines off the boundary to move closer to the boundary, assuming that η increases off the boundary. The effect of the P function will be to alter the angle at which the ξ-lines intersect the boundary, since the points on the boundary are fixed, with the ξ-lines tending to lean in the direction of decreasing ξ. If the boundary is such that η decreases off the boundary then the amplitudes in the Q function must be negative to achieve attraction to the boundary. In any case, the amplitudes a_i cause the effects to occur all along the boundary, while the effects of the amplitudes b_i occur only near selected points on the boundary.

If the attraction line and/or the attraction points are in the field, rather than on a boundary, then the attraction is not to a fixed line or point in space, since the attraction line or points are themselves solutions of the system of equations, the functions P and Q being functions of the variables ξ and η. It is, of course, also possible to take these control functions as functions of x and y, instead of ξ and η, and achieve attraction to fixed lines and/or points in the physical field. This case becomes somewhat more complicated, since it must be ensured that coordinate lines are not attracted parallel to themselves, and its discussion follows in a later section.

C. Control Functions for Certain Spacings

For certain simple geometries it is possible to integrate (8) analytically for appropriately selected forms of the control functions, and thus to determine the control functions required to produce a certain line spacing. In this regard consider the case of two concentric circular boundaries of radii r_1 and r_2, with $r_2 > r_1$.

With $\eta = 1$ on the inner boundary, $\eta = J$ on the outer boundary, and ξ varying monotonically from 1 to I around these boundaries, a solution of (8) can be given in the form

$$x = r(\eta) \cos \left[2\pi \left(\frac{\xi-1}{I-1}\right)\right] \tag{11a}$$

$$y = r(\eta) \sin \left[2\pi \left(\frac{\xi-1}{I-1}\right)\right] \tag{11b}$$

Substitution of these expressions into the equations of (8) with $P(\xi,\eta) = 0$ yields

$$\frac{r''}{r'} - \frac{r'}{r} + r'^2 Q = 0 \tag{12}$$

This can be made a perfect differential by taking the control function Q to be of the form (following the direction of Ref. 7)

$$Q = -\frac{f''(\eta)}{f'(\eta)} \frac{1}{r'^2}$$

where the minus sign has been introduced merely for convenience. Since $\frac{1}{r'^2}$ is equal to $\frac{\gamma}{J^2}$ for the solution given by (11), this form of Q suggests taking Q to be of the form

$$Q = -\frac{\gamma}{J^2} \frac{f''(\eta)}{f'(\eta)} \tag{13}$$

Substitution of (13) into (12) yields

$$\frac{r''}{r'} - \frac{r''}{r} - \frac{f''}{f'} = 0 \tag{14}$$

which can be integrated twice to yield

$$r(\eta) = c_2 e^{c_1 f(\eta)}$$

The constants of integration may be evaluated from the boundary conditions, $r(1) = r_1$, $r(J) = r_2$, so that

$$r(\eta) = r_1 \, (r_2/r_1)^{\left[\frac{f(\eta) - f(1)}{f(J) - f(1)}\right]} \tag{15}$$

This equation may then be solved for $f(\eta)$ to yield

$$\frac{f(\eta) - f(1)}{f(J) - f(1)} = \frac{\ln\left[\frac{r(\eta)}{r_1}\right]}{\ln\left[\frac{r_2}{r_1}\right]} \tag{16}$$

If the distance from the body to the Nth η-line is specified to be r_N, the following equation must be satisfied:

$$\frac{f(N) - f(1)}{f(J) - f(1)} = \frac{\ln\left[\frac{r_N}{r_1}\right]}{\ln\left[\frac{r_2}{r_1}\right]} \tag{17}$$

It should be noted that the form of $f(\eta)$ is still arbitrary, subject to (17).

Alternatively, to set r' at the inner boundary, $\eta = 1$, we have, upon differentiation of (15) with respect to η and subsequent evaluation at $\eta = 1$, that $f(\eta)$ must satisfy

$$\frac{f'(1)}{f(J) - f(1)} = \frac{\dfrac{r'(1)}{r_1}}{\ln(\dfrac{r_2}{r_1})} \tag{18}$$

The two derivatives appearing in the truncation error of first derivatives, as given in last equation in section II, are, from repeated differentiation of (15),

$$r' = \frac{\ln(\dfrac{r_2}{r_1})}{f(J) - f(1)} [rf'(\eta)] \tag{19a}$$

$$r'' = \frac{\ln(\dfrac{r_2}{r_1})}{f(J) - f(1)} [r'f'(\eta) + rf''(\eta) \tag{19b}$$

Thus if a function $f(\eta)$ with a free parameter is selected, (17) may be used to determine the parameter in the function such that the Nth η-line lies at a specified distance, $r_N - r_1$, from the inner boundary. Alternatively, the free parameter may be determined by (18) such that the spacing at the boundary is set by specification of r' there. The derivatives in the truncation error terms may then be calculated from (19). With the function $f(\eta)$ determined, the control function Q is then given by (13).

For example, with the function (Ref. 7)

$$f(\eta) \equiv \eta K^{\eta-1}$$

where K is a free parameter, we have, by (17), that K must be the solution of the nonlinear equation

$$\frac{NK^{N-1} - 1}{JK^{J-1} - 1} = \frac{\ln(\dfrac{r_N}{r_1})}{\ln(\dfrac{r_2}{r_1})} \tag{20}$$

to set the Nth η-line at r_N. Alternatively, the value of K required to set r' to a specified value at the inner boundary is determined by (18) as the solution of the nonlinear equation

$$\frac{1 + \ln K}{JK^{J-1} - 1} = \frac{\frac{r'(1)}{r_1}}{\ln(\frac{r_2}{r_1})} \tag{21}$$

For this function, the derivatives appearing in the truncation error term, (19), are given by

$$f'(\eta) = (1 + \eta \ln K)K^{\eta-1} \tag{22a}$$

$$f''(\eta) = (2 + \eta \ln K)(\ln K)K^{\eta-1} \tag{22b}$$

The control function Q is given by (13) as

$$Q = -\frac{1}{J^2}(\frac{2 + \eta \ln K}{1 + \eta \ln K})\ln K \tag{23}$$

It can be shown by consideration of the ratios of successive derivatives that the higher derivatives of this function are progressively decreasing if K is in the range

$$0 < \ln K < \frac{1}{2}(\sqrt{5} - 1) \tag{24}$$

Since the left side of (21) is a decreasing function of K for positive K, the smallest value of the spacing at the boundary, r'(1), that can be achieved while maintaining progressively decreasing higher derivatives of $f(\eta)$ occurs with K at the upper limit of the inequality (24), viz

$$r'(1)_{\min} = r_1 \ln(\frac{r_2}{r_1}) \frac{\frac{1}{2}(1 + \sqrt{5})}{J \exp[\frac{1}{2}(\sqrt{5} - 1)(J - 1)] - 1} \tag{25}$$

It is not reasonable to use smaller values of $r'(1)$ since the progressively increasing higher derivatives of $f(\eta)$ will result in significant truncation error introduced by the coordinate system.

Another choice of $f(\eta)$ might be

$$f(\eta) = \sinh\,[K(\eta - 1)] \tag{26}$$

for which, from (17), the Nth line occurs at r_N for K given by the solution of the equation

$$\frac{\sinh\,[K(N - 1)]}{\sinh\,[K(J - 1)]} = \frac{\ln(\frac{r_N}{r_1})}{\ln(\frac{r_2}{r_1})}$$

or the spacing at the inner boundary is $r'(1)$ for K given by (18):

$$\frac{K}{\sinh\,[K(J - 1)]} = \frac{\frac{r'(1)}{r_1}}{\ln(\frac{r_2}{r_1})} \tag{27}$$

The first two derivatives and the control function are given by

$$f'(\eta) = K\cosh[K(\eta - 1)] \tag{28a}$$

$$f''(\eta) = K^2\sinh\,[K(\eta - 1)] \tag{28b}$$

$$Q = -\frac{1}{J^2}\,K\tanh\,[K(\eta - 1)] \tag{29}$$

In this case progressively decreasing higher derivatives occur for K in the range $0 < K < 1$, so that the smallest practical spacing at the inner boundary is

$$r'(1)_{min} = \frac{r_1 \ln(\frac{r_2}{r_1})}{\sinh(J - 1)}$$

The control function for the spacing distribution of Roberts, Ref. 8, can be determined in the same manner as follows. With the notation of Ref. 8 adjusted so that the boundaries occur as used above, we have

$$r(\eta) = r_2 + \frac{G(\eta) - 1}{G(\eta) + 1}(1 - \frac{r_1}{r_2})b \tag{30}$$

with

$$G(\eta) = (\frac{b + r_2}{b - r_2})^{(\frac{\eta - J}{J - 1})}$$

with b a free parameter.

Although the form of $f(\eta)$ could be extracted by substitution of (30) into (16), it is simpler to determine the parameter b from either $r(N) = r_N$, or for a specified value of $r'(1)$. The derivatives are

$$r'(\eta) = \frac{2}{J - 1} \ln(\frac{b + r_2}{b - r_2}) \frac{G}{(G + 1)^2} (1 - \frac{r_1}{r_2})b \tag{31a}$$

$$r''(\eta) = 2(\frac{1}{J - 1})^2 \ln^2 (\frac{b + r_2}{b - r_2}) \frac{G(1 - G)}{(G + 1)^3} (1 - \frac{r_1}{r_2})b \tag{31b}$$

The control function Q is then given by (13) and (14) as

$$Q = - \frac{1}{J^2}(\frac{r''}{r'} - \frac{r'}{r}) \tag{32}$$

with r, r', and r" to be substituted from (30) and (31).

Finally, another type of function is a patched function using different functions near and away from the inner boundary to achieve a group of closely spaced lines near the inner boundary with fairly rapid expansion outside this inner group. This is done as follows: Let the spacing of the inner group be such that the same change in velocity would occur between each two lines for a velocity distribution given by u(r). To do this, invert the velocity function such that r = r(u), and then take

$$u(\eta) = \frac{\eta - 1}{N - 1} u_N \qquad 1 \leq \eta \leq N$$

when u_N is the velocity at the edge of the inner group of lines. Then

$$r(\eta) = r(u = \frac{\eta - 1}{N - 1} u_N) \qquad 1 \leq \eta \leq N$$

From this function all the derivatives and the control function may be calculated, the latter being determined by (13) and (14).

Now outside the inner group of lines, i.e., for $N \leq \eta \leq J$, let $r(\eta)$ be a quartic polynomial:

$$r(\eta) = r'_N(\eta - N) + \frac{1}{2}r''_N(\eta - N)^2 + \frac{1}{6}r'''_N(\eta - N)^3$$
$$+ a(\eta - N)^4 + r_N \qquad N \leq \eta \leq J$$

where the three derivatives at $\eta = N$ are determined from the derivatives of the inner function, all being evaluated at $\eta = N$. The final parameter, a, is determined such that $r(J) = r_2$. Thus

$$a = \frac{(r_J - r_N) - r'_N(J - N) - \frac{1}{2}r''_N(J - N)^2 - \frac{1}{6}r'''(J - N)^3}{(J - N)^4}$$

The outer control function is then determined from (13) and (14). This composite control function has only one continuous derivative, and thus could possibly lead to truncation error introduced by the coordinate system.

It is, of course, also possible to integrate the coordinate equation (8) for the one-dimensional case. In that case the control function Q is given by

$$Q = -\frac{\gamma}{J^2}\frac{r''}{r'} = -\frac{\gamma}{J^2}\frac{f''(\eta)}{f'(\eta)} \tag{33}$$

and

$$r(\eta) = r_2 + (r_2 - r_1)\left\{\frac{f(\eta) - f(1)}{f(J) - f(1)}\right\} \tag{34}$$

All the other steps follow in analogy with the two-dimensional case.

Now, although the two-dimensional case given above applies only to concentric circular boundaries, the effect of using the same control functions for the general case will be qualitatively the same, with even closer spacing near inner boundary with stronger curvature. Thus the control functions derived in the above manner can be expected to produce the type of spacing desired in general applications. A version of the TOMCAT code incorporating several of these functions has been written and has been used to produce coordinate systems for airfoils with the spacing at the airfoil set at $\frac{0.01}{\sqrt{R}}$ automatically through (18). An example is shown in Fig. 4, using the function above (20). Other examples are given in Ref. 9.

D. Revised Generating System

The form of the control function Q taken in (13) naturally leads to the idea of replacing the original elliptic system, (7), with the system

$$\nabla^2\xi = (\xi_x^{\ 2} + \xi_y^{\ 2})\ P\ (\xi,\eta) \tag{35a}$$

$$\nabla^2\eta = (\eta_x^{\ 2} + \eta_y^{\ 2})\ Q\ (\xi,\eta) \tag{35b}$$

since the terms multiplying P and Q here are, respectively, equal to $\frac{\alpha}{J^2}$ and $\frac{\gamma}{J^2}$. With this system the transformed equations are

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -(\alpha P x_\xi + \gamma Q x_\eta) \tag{36a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -(\alpha P y_\xi + \gamma Q y_\eta) \tag{36b}$$

This form has also been given by Shanks and Thompson, Ref. 10, and by Thomas and Middlecoff, Ref. 11. This form has now been adopted in the latest version of the TOMCAT code.

The exponential forms of the functions P and Q, and the discussion given therewith above, are still applicable with this system. Appropriate values of the attraction amplitudes are several orders of magnitude smaller with this new system because of the relatively large values attained by the terms multiplying P and Q for small Jacobians.

Finally, it is useful to solve (36) simultaneously to display P and Q explicitly as

$$P = -\frac{1}{\alpha J}(y_\eta Dx - x_\eta Dy) \tag{37a}$$

$$Q = \frac{1}{\gamma J}(y_\xi Dx - x_\xi Dy) \tag{37b}$$

with

$$Dx = \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} \tag{38a}$$

$$Dy = \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} \tag{38b}$$

With (37) the control functions required to produce any specified solution $x(\xi,\eta)$, $y(\xi,\eta)$ could be calculated. Although such a procedure is normally of only academic interest, since the solution $x(\xi,\eta)$, $y(\xi,\eta)$ is yet to be determined, it might be useful in some cases to determine

P and Q from (37) for some approximate solution generated, say, by simple interpolation from the boundaries, and then to use smoothed values of these functions as the control functions for the actual solution. Although the approximate solution might have lacked continuity of derivatives, the actual solution determined by solving the elliptic system with the smoothed control functions will have continuous derivatives, while following generally the form of the approximate solution.

E. Control Functions for Near Orthogonality at Boundary

Another example of the usefulness of (37) is as follows. The solution for the concentric circle case can be generalized slightly to include variable spacing of points along the boundaries by taking, instead of (11),

$$x = r(\eta) \cos \left[2\pi \frac{g(\xi) - g(1)}{g(I) - g(1)}\right] \tag{39a}$$

$$y = r(\eta) \sin \left[2\pi \frac{g(\xi) - g(1)}{g(I) - g(1)}\right] \tag{39b}$$

Substitution of these functions in (37) then results in

$$P = -\frac{g''}{g'} \tag{40a}$$

$$Q = \frac{r'}{r} - \frac{r''}{r'} \tag{40b}$$

The second of these is the same as (13), using (14) and considering the above re-definition of Q, and was used above to generate the control function Q.

With g(η) determined by the boundary point spacing, the control function P given here will maintain the ξ-lines as radial lines, i.e., normal to the circular boundaries. Note that arc length along the circular boundary is given by

$$s(\xi) = 2\pi \frac{g(\xi) - g(1)}{g(I) - g(1)} r \tag{41}$$

so that the function $g(\xi)$ may be related to arc length by

$$g(\xi) = g(1) + \frac{g(I) - g(1)}{2\pi r} s(\xi) \tag{42}$$

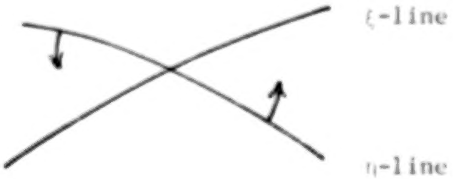
Thus (40a) can be rewritten in terms of arc length as

$$P = -\frac{s''}{s'} \tag{43}$$

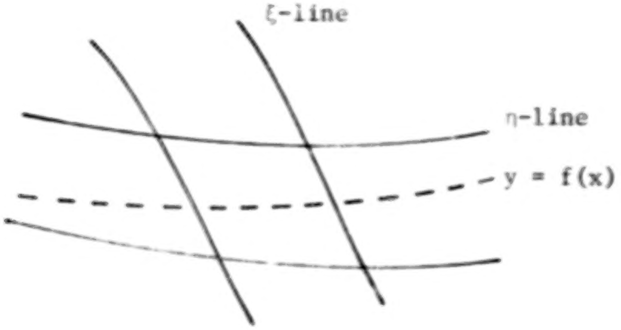
As discussed above for the control function Q, this idea can be carried over to the case of general boundaries to produce the same effect qualitatively. Thus in the general case, the control function P could be determined at each boundary from (43), and then values of P in the field could be taken from linear interpolation between the values at corresponding boundary points.

F. Attraction to Fixed Lines in Physical Space

As mentioned above, the attraction of coordinate lines to fixed lines and/or points in physical space, rather than to floating coordinate lines and/or points, requires further consideration. Recall that in the above discussion, η-lines are attracted to other η-lines, and ξ-lines are attracted to other ξ-lines. It is unreasonable, of course, to attempt to attract η-lines to ξ-lines, since that would have the effect of collapsing the coordinate system:



When, however, the attraction is to be to certain fixed lines in x-y space, defined by curves $y = f(x)$, care must be exercised to avoid attempting to attract η or ξ lines to specified curves that cut the η or ξ lines at large angles. Thus, in the figure below:



it is unreasonable to attract ξ lines to the curve $f(x)$, while it is natural to attract the η-lines to $f(x)$.

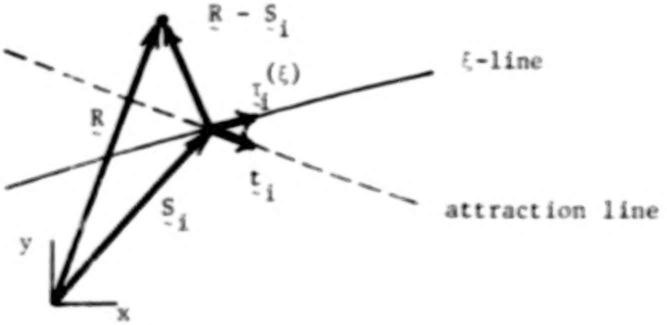
However in the general situation, the specified line $f(x)$ will not necessarily be aligned with either a ξ or η line along its entire length. Since it is unreasonable to attract a line parallel to itself, some provision is necessary to decrease the attraction to zero as the angle between the coordinate line and the given line $f(x)$ goes to zero. This can be accomplished by multiplying the attraction function by the cosine of the angle between the coordinate line and the line $f(x)$. It is also necessary to change the sign on the attraction function on either side of the line $f(x)$. This can be done by multiplying by the sine of the angle between the line $f(x)$ and the vector to the point on coordinate line.

These two purposes can be accomplished as follows. Let a general point (x, y) be located by the vector $\underline{R}(x, y)$, and let the attraction line $y = f(x)$ be specified by the collection of points $\underline{S}(x_i, y_i)$, $i = 1, 2, \text{--}, n$. Let the unit tangent to the attraction line be $\underline{t}(x_i, y_i)$, and the unit tangent to a ξ-line be $\underline{\tau}^{(\xi)}$. Then the sine and cosine of the angle between the ξ-line and the attraction line may be written as

$$\text{sine} = \frac{[\underline{t}_i \times (\underline{R} - \underline{S}_i)] \cdot \underline{k}}{|\underline{R} - \underline{S}_i|}$$

$$\text{cosine} = \underline{t}_i \cdot \underline{\tau}_i^{(\xi)}$$

where $\underline{k}$ is the unit vector normal to the two dimensional plane. These relations are evident from the figure



The control function P(x, y) may then be logically taken as

$$P(x,y) = -\sum_{i=1}^{n} a_i(\underline{t}_i \cdot \underline{\tau}_i^{(\xi)}) \frac{[\underline{t}_i \times (\underline{R} - \underline{S}_i)] \cdot \underline{k}}{|\underline{R} - \underline{S}_i|} \exp(-d_i |\underline{R} - \underline{S}_i|) \tag{44a}$$

with the analagous form for Q:

$$Q(x,y) = -\sum_{i=1}^{n} a_i(\underline{t}_i \cdot \underline{\tau}_i^{(\eta)}) \frac{[\underline{t}_i \times (\underline{R} - \underline{S}_i)] \cdot \underline{k}}{|\underline{R} - \underline{S}_i|} \exp(-d_i |\underline{R} - \underline{S}_i|) \tag{44b}$$

These functions depend on x and y through both $\underline{R}$ and $\underline{\tau}^{(\xi)}$ or $\underline{\tau}^{(\eta)}$ and thus must be recalculated at each point as the iterative solution of (36) proceeds. This form of coordinate control will therefore be more expressive than that based on attraction to other coordinate lines.

There is no real distinction between "line" and "point" attraction with this type of attraction. "Line" attraction here is simply attraction to a group of points that form a line f(x). If line attraction is specified, then the tangent to the line f(x) is computed from the adjacent points on the line. If point attraction is specified, then the "tangent" must be input for each point.

The tangents to the coordinate lines are computed from

$$\underset{\sim}{t}^{(\xi)} = \frac{1}{\sqrt{\alpha}}(\underset{\sim}{i}x_\eta + \underset{\sim}{j}y_\eta) \tag{45a}$$

$$\underset{\sim}{t}^{(\eta)} = \frac{1}{\sqrt{\gamma}}(\underset{\sim}{i}x_\xi + \underset{\sim}{j}y_\xi) \tag{45b}$$

G. Point Distribution on Boundary According to Curvature

One final technique to mention concerns the placement of points along the boundary according to the local boundary curvature. Let a boundary curve be described by the function $y = f(x)$. Then if s is arc length along the boundary we have

$$\frac{ds}{dx} = \sqrt{1 + f'^2}$$

Now take the rate of change of arc length with the curvilinear coordinate, ξ, along the boundary to be exponentially dependent on the local radius of curvature, r, of the boundary. Thus let

$$\frac{ds}{d\xi} = 1 - e^{-br}$$

where b is a free parameter. This function causes the arc length to change slowly with ξ where the curvature is large.

Then

$$\frac{d\xi}{dx} = \frac{ds}{dx}\bigg/\frac{ds}{d\xi} = \frac{\sqrt{1 + f'^2}}{1 - e^{-br}}$$

Since f and r are known at each x, a normalized $\xi(x)$ may be determined from

$$\xi(x) = 1 + \frac{\displaystyle\int_0^x \frac{\sqrt{1 + f'^2}}{1 - e^{-br}}\, dx'}{\displaystyle\int_0^1 \frac{\sqrt{1 + f'^2}}{1 - e^{-br}}\, dx'}\, I \tag{46}$$

assuming x is normalized to vary from 0 to 1 and ξ to vary from 1 to I. The quadrature may be taken numerically if necessary.

Then for I number of ξ-points $\xi = 1, 2, --, I$ on the boundary, the corresponding values of x can be determined by inversion of $\xi(x)$, done by interpolation of tabular values if necessary. The arc length between each of these points can then be calculated and the value of the free parameter b can be adjusted iteratively to produce, say, a specified maximum arc spacing along the boundary, or perhaps, to match a specified arc spacing at either end or, for that matter, at any given point. In application to airfoils, this procedure is applied to the upper surface with b chosen to match a specified maximum arc spacing. A separate application is then made to the lower surface with b there being chosen to match the arc spacing adjacent to the leading edge on the upper surface.

This procedure produces a smooth point distribution on the boundary, with points concentrated in regions of large curvature, yet free of the rapid spacing changes that lead to coordinate-system-introduced truncation error of the type discussed in an earlier section.

IV. SOME RECENT APPLICATIONS OF COORDINATE SYSTEMS

In addition to extensive application to airfoils, as illustrated in Fig. 4, in which the transformed plane is an empty rectangle, some more general configurations have recently been treated using a transformed plane that contains rectangular voids as discussed in Ref. 12. For example, a coordinate system used in a simulation of a nuclear reactor cooling system is shown in Fig. 5, taken from Ref. 13, and systems for Charleston harbor (Ref. 14) and a portion of Lake Ponchatrain are shown in Figs. 6 and 7.

V. CONCLUSION

Control of the spacing of coordinate lines so as to resolve large gradients in numerical solution of partial differential equations continues to be of paramount importance. Research has provided some means of control and of error estimation. The experience gained thus far has indicated the versatility of the coordinate systems generated from elliptic systems and the possibility of optimization of such systems in adaptation to the nature of particular partial differential systems and boundary configuration.

REFERENCES

1. F. G. Blottner and P. J. Roache, "Nonuniform Mesh Systems," Journal of Computational Physics, 8 (1971), 498-499.

2. H. J. Crowder and C. Dalton, "Errors in the Use of Nonuniform Mesh Systems," Journal of Computational Physics, 7 (1971), 32-45.

3. E. Kalnay de Rivas, "On the Use of Nonuniform Grids in Finite-Difference Equations," Journal of Computational Physics, 10 (1972), 202-210.

4. B. L. Pierson and P. Kutler, "Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics," AIAA Paper 79-0272, (1979).

5. C. W. Mastin and J. F. Thompson, "Errors in Finite-Difference Computations on Curvilinear Coordinate Systems, MSSU-EIRS-ASE-80-4, Department of Aerospace Engineering, Mississippi State University, (1980).

6. J. F. Thompson, F. C. Thames, C. W. Mastin, "TOMCAT" - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 24 (1977), 274.

7. Z. U. A. Warsi, J. F. Thompson, "Machine Solutions of Partial Differential Equations in the Numerically Generated Coordinate Systems," Report MSSU-EIRS-ASE-77-1, Engineering and Industrial Research Station, Mississippi State University, (1976).

8. G. O. Roberts, "Computational Meshes for Boundary Layer Problems," Proc. of the 2nd Int. Conf. on Numerical Methods in Fluid Mechanics, Lecture Notes in Physics, 8 (1970), 171.

9. J. F. Thompson and D. S. Thompson, "Control of Coordinate Line Spacing for Numerical Simulation of Flow About Airfoils," MSSU-EIRS-ASE-80-5, Department of Aerospace Engineering, Mississippi State University, (1980).

10. S. P. Shanks and J. F. Thompson, "Numerical Solution of the Navier-Stokes Equation for 2D Hydrofoils In or Below a Free Surface." Proc. of the 2nd International Conference on Numerical Ship Hydrodynamics, Berkeley, 1977.

11. P. D. Thomas and J. F. Middlecoff, "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations." AIAA Journal, 13, 932, 1980.

12. J. F. Thompson, "Numerical Solution of Flow Problems Using Body-Fitted Coordinate Systems," Lecture Series in Computational Fluid Dynamics, von Karman Inst. for Fluid Dynamics, Belgium, (1978).

13. W. T. Sha, B. C-J Chen, Y. S. Cha, S. P. Vanka, R. C. Schmitt, J. F. Thompson, and M. L. Doria, "Benchmark Rod-Bundle Thermal-Hydraulic Analysis Using Boundary Fitted Coordinates," presented at 1979 Winter Meeting of the American Nuclear Society, San Francisco, (1979).

14. B. H. Johnson and J. F. Thompson, "A Discussion of Boundary-Fitted Coordinate Systems and Their Applicability to the Numerical Modeling of Hydraulic Problems," Misc. Paper H-78-9, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Miss., (1978).

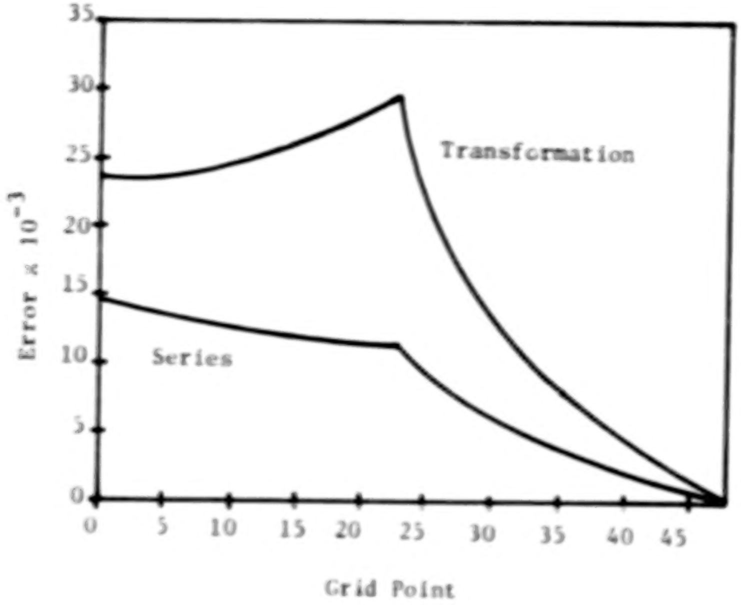


Figure 1.- Error at grid points on the coordinate line $1 \leq x \leq 10$, $y = 0$.

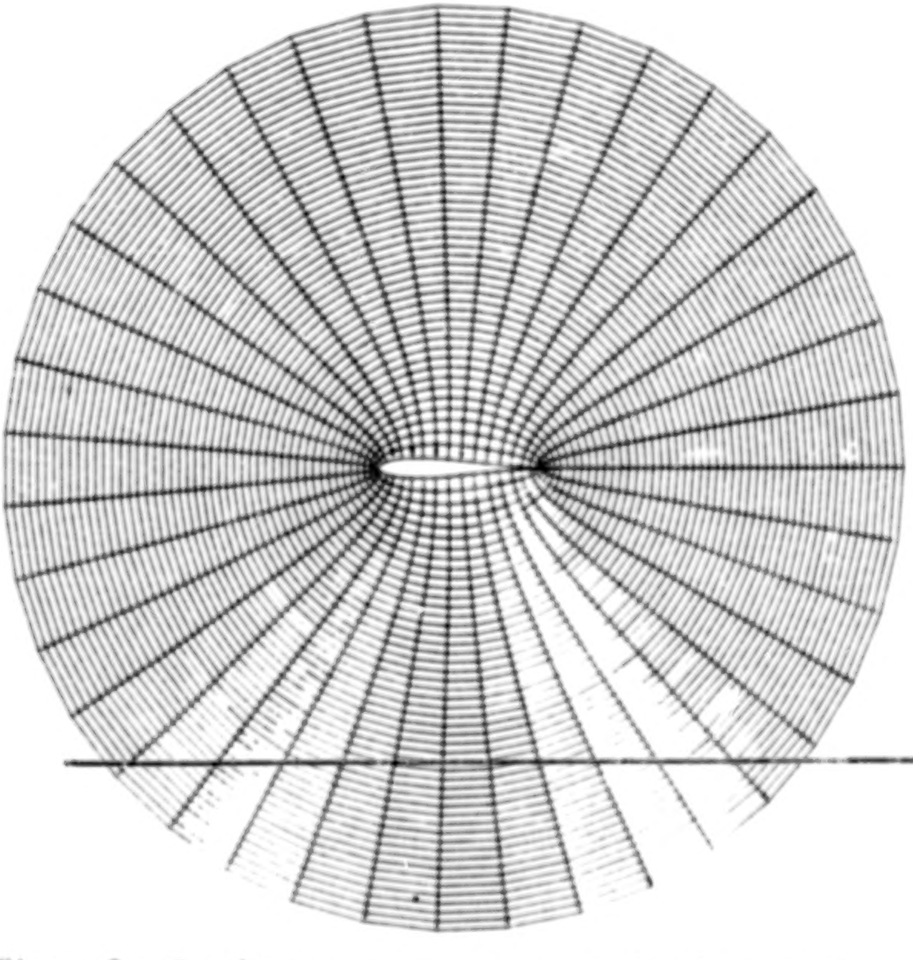


Figure 2.- Coordinate system about a Joukowski airfoil.

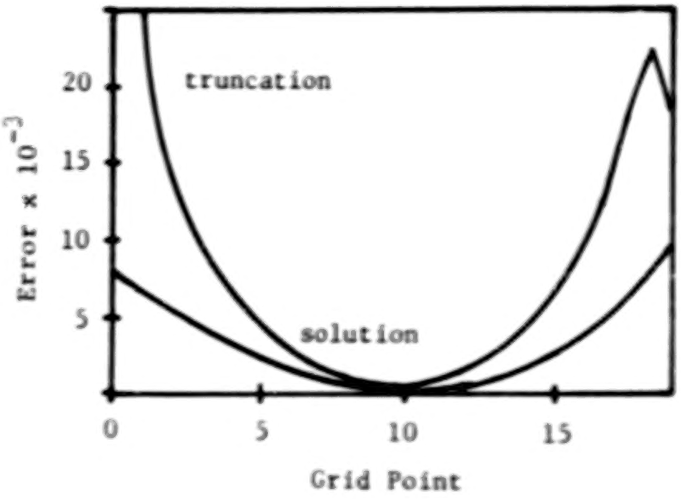


Figure 3.- Comparison of estimated truncation error in the Laplacian with the error in the solution of Laplace's equation.

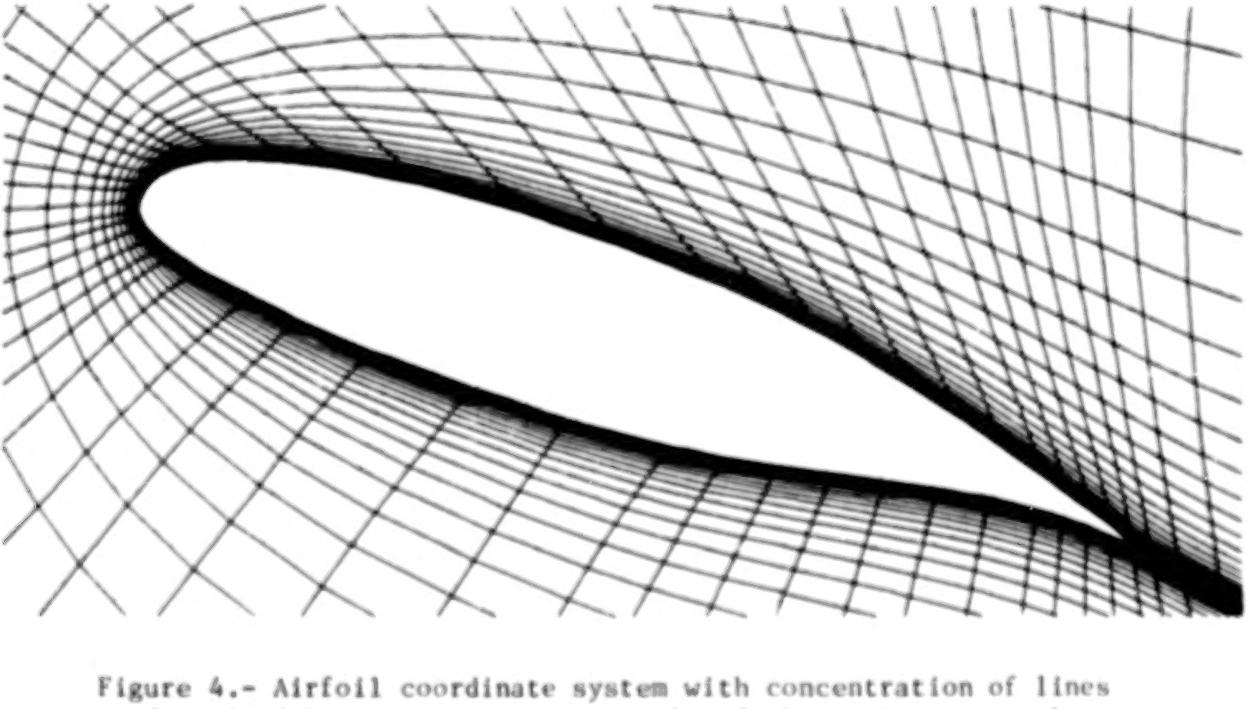
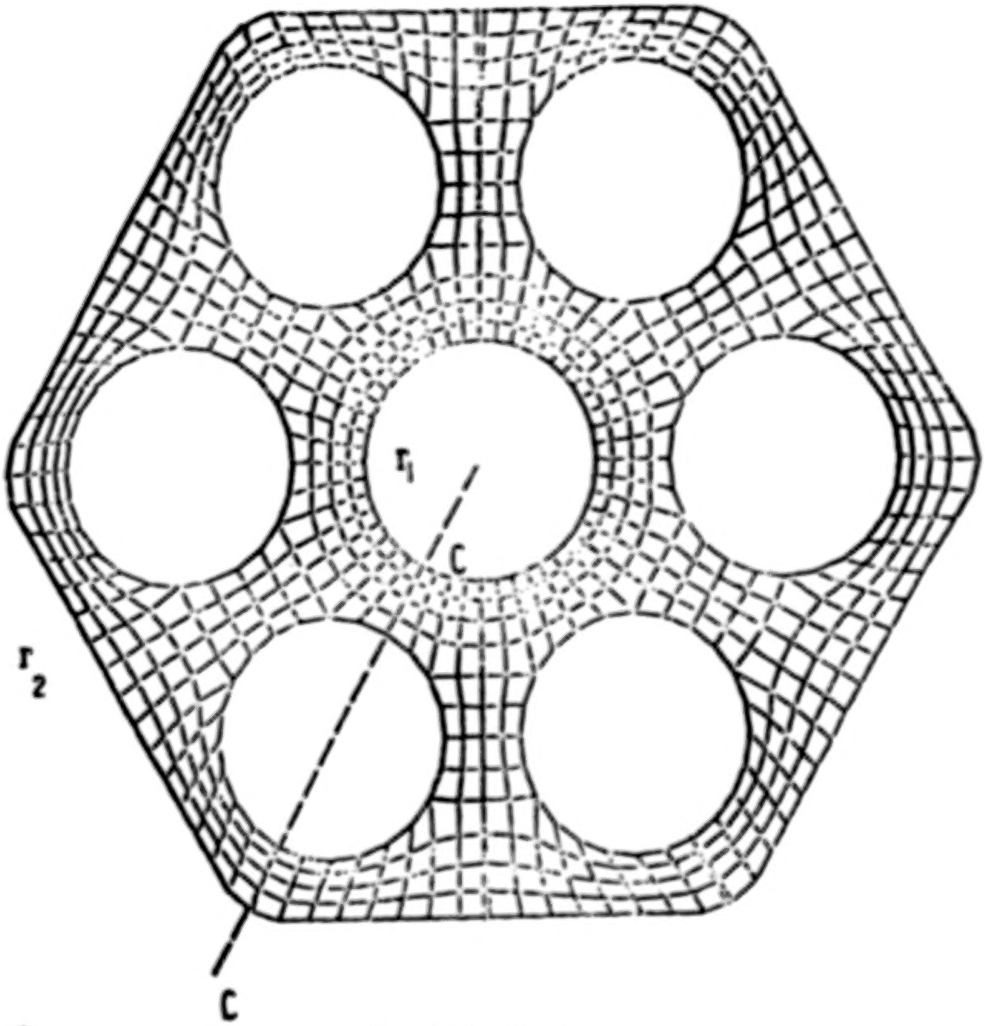
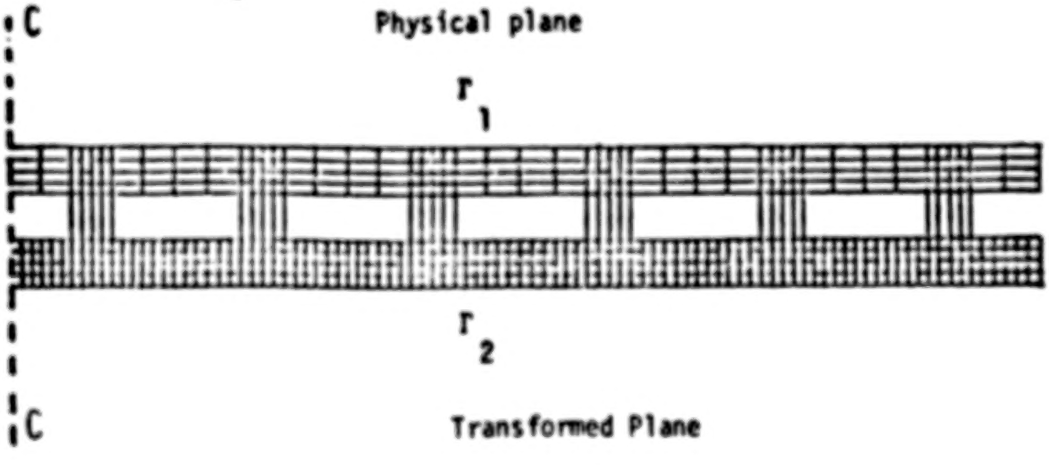


Figure 4.- Airfoil coordinate system with concentration of lines (supplied by Dr. Krishna Devarayalu of the Boeing Company).


Γ_1
C
Γ_2
C

Physical plane


C
Γ_1
Γ_2
C

Transformed Plane

Figure 5.- Nuclear rod-bundle coordinate system (Ref. 13).

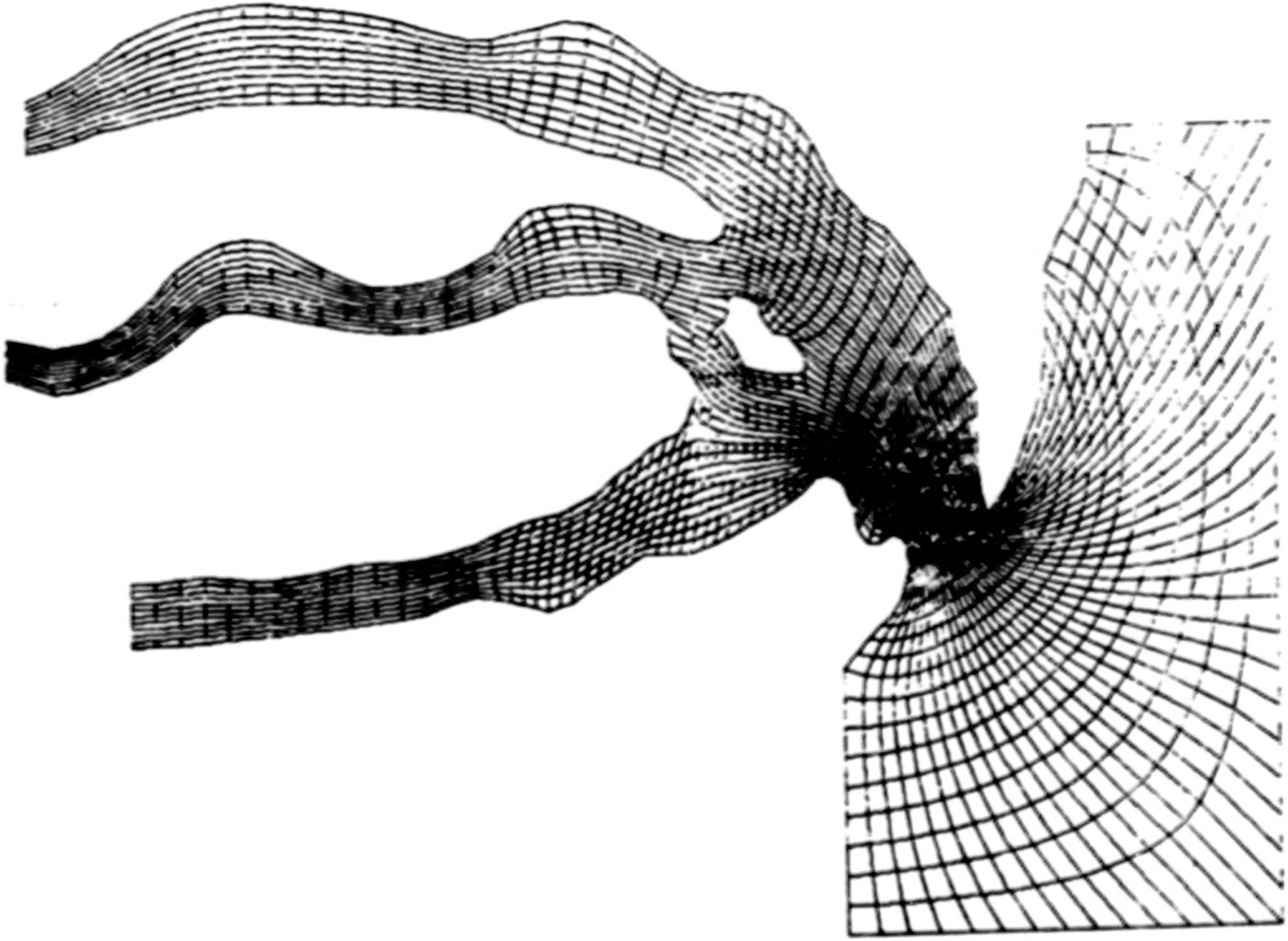


Figure 6.- Coordinate system for Charleston Harbor (Ref. 14).

BLANK PAGE

BLANK PAGE

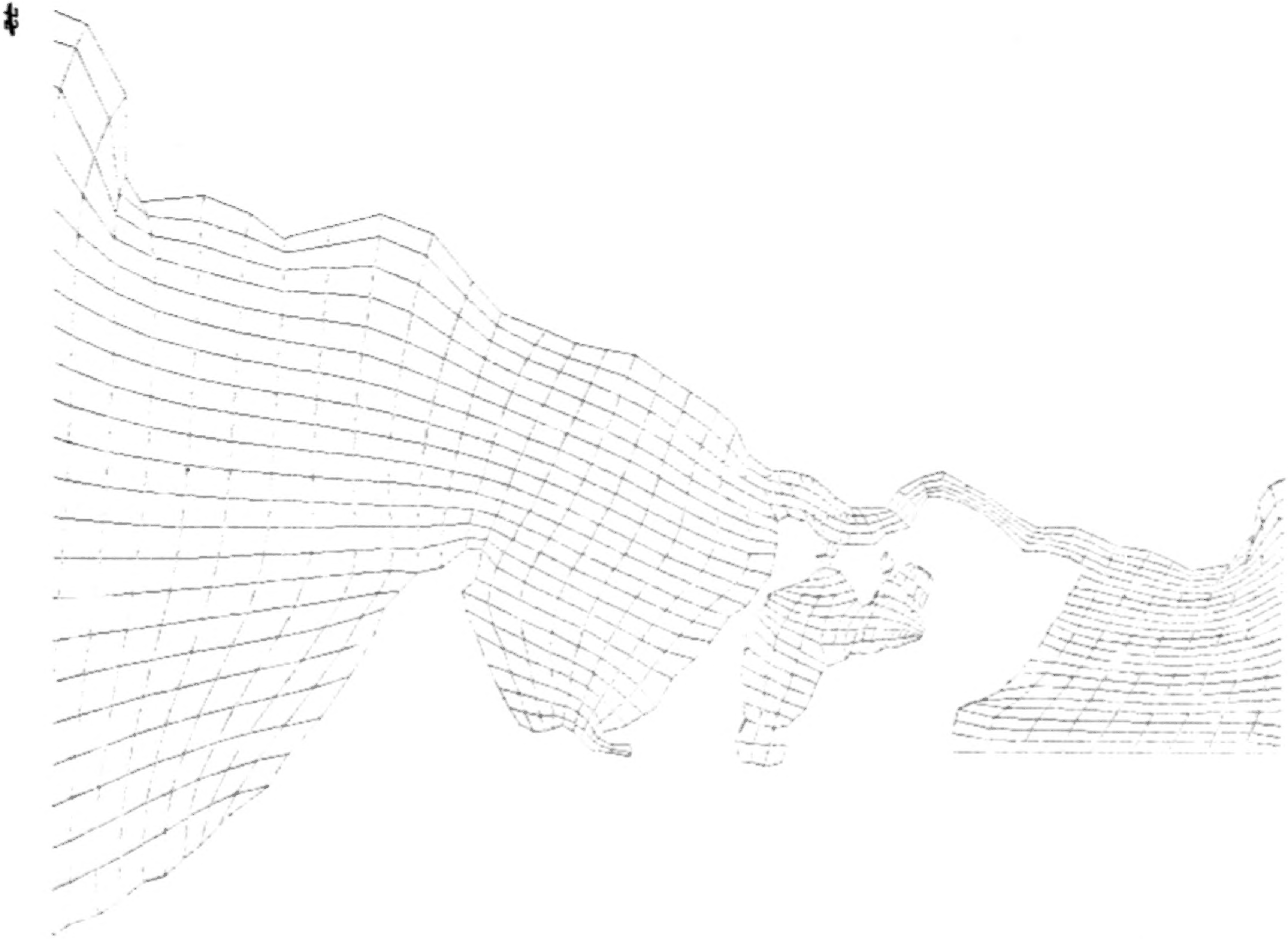


Figure 7.- Coordinate system for a portion of Lake Ponchatrain.

MESH GENERATION USING ALGEBRAIC TECHNIQUES

Peter R. Eiseman
Institute for Computer Applications in
Science and Engineering, USRA

and

Robert E. Smith
NASA Langley Research Center

Coordinate transformations are powerful tools for the solution of the partial differential equations which describe physical phenomena. The use of transformations leads to well ordered discretizations of the physical domain and thereby renders a simplification in a numerical solution process. The discretization is constrained by the underlying physics, the problem geometry and the topology of the region where the solution is to be obtained. The constraints can be stated in geometric terms. In particular they can be categorized as boundary constraints, uniformity constraints, and internal constraints. Boundary constraints include: the basic geometry of solid objects, the transmissive junctures between and around solid objects, the choice of representation for the boundaries, the angles at which transverse coordinate curves intersect boundaries, and the rate of entry for such coordinate curves. Uniformity constraints are applied to either local or global distributions of coordinate curves or points to form a basis from which the curves or points can be redistributed.

This may be based on an a priori specification of a distribution function or on a solution adaptive approach. In either case, the redistribution must not be distorted by the underlying transformation. Internal constraints are applicable when an interior shape or interior mesh structure is to be smoothly embedded within a global mesh to simplify the simulation of physical processes in the given region.

Algebraic mesh generation techniques are highly advantageous for meeting the constraints described above. Algebraic techniques provide exact control of the mesh properties necessary to satisfy the given constraints. Although other methods have been developed which provide some degree of control, the level of control is not in general sufficient to satisfy certain of the constraints. For example, the smooth embedding of a Cartesian mesh within a global mesh structure cannot be readily constructed with the application of differential equation techniques. Also, three dimensional meshes are not in general readily obtained with non-algebraic techniques. On the other hand, algebraic techniques require more complex specification of the data to assemble a mesh. The purpose of this paper is to present an overview of algebraic techniques for mesh generation and set forth the underlying concepts which have been successful. Both two- and three-dimensional domains are considered.

The Multi-Surface Transformation

When curvilinear coordinates are employed in the numerical solution of a boundary value problem, constraints must often be placed upon the coordinates, in addition to the basic requirement that the bounding surfaces are coordinate surfaces of one or more coordinate systems. The locations of the constraints can occur anywhere in the problem domain. On the boundaries, a particular pointwise distribution may be needed; in regions near boundaries, a particular coordinate form may be advantageous; and away from the boundaries, an internal coordinate specification may also be required. Typically, the constraints will arise either to resolve regions with large solution gradients or to cause some simplification in the problem formulation and solution.

In conjunction with the demand for constraints, the general multi-surface transformation [1] will be examined. The multi-surface transformation is a method for coordinate generation between an inner bounding surface $\vec{P}_1$ and outer bounding surface $\vec{P}_N$. To establish a particular distribution of mesh points on each bounding surface, a common parameterization $\vec{t}$ is chosen for each surface. This is equivalent to a coordinate description of the surfaces which yields the desired surface mesh when the parametric components of $\vec{t}$ are given a uniform discretization. With the parametric description, the inner and outer bounding surfaces are denoted by $\vec{P}_1(\vec{t})$ and $\vec{P}_N(\vec{t})$ respectively. In continuation, parameterized intermediate surfaces $\vec{P}_2(\vec{t}),\ldots,\vec{P}_{N-1}(\vec{t})$ are introduced so that they can be used as controls over the internal form of the coordinates. The intermediate surfaces are not coordinate surfaces but, instead, are surfaces which are used to establish a vector field that is composed of tangent vectors to the coordinate curves spanning the coordinate system to connect bounding surfaces. It is also assumed that the collection of surfaces $\vec{P}_1(\vec{t})$, $\vec{P}_2(\vec{t}),\ldots,\vec{P}_N(\vec{t})$ is ordered from bounding surface to bounding surface. An illustration is given in Fig. 1. For a fixed

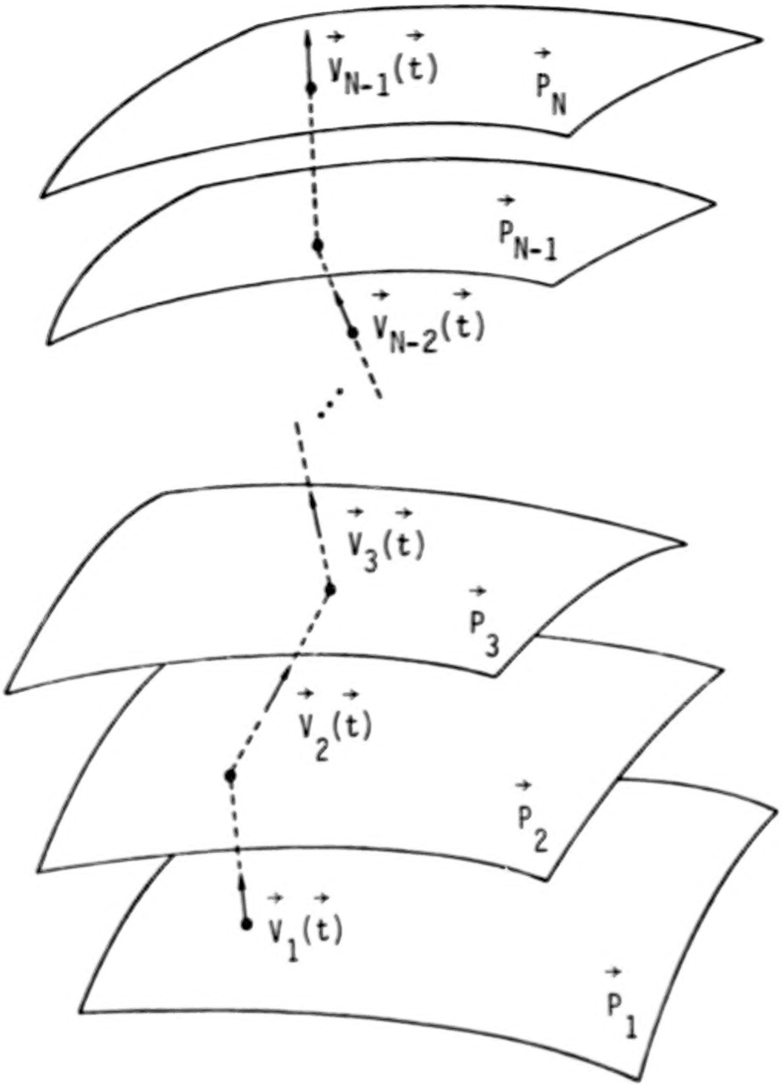

$\vec{V}_{N-1}(\vec{t})$
$\vec{P}_N$
$\vec{P}_{N-1}$
$\vec{V}_{N-2}(\vec{t})$
$\vec{V}_3(\vec{t})$
$\vec{P}_3$
$\vec{V}_2(\vec{t})$
$\vec{P}_2$
$\vec{V}_1(\vec{t})$
$\vec{P}_1$

Fig. 1 - A piecewise linear curve and its tangent field.

parameter value $\vec{t}$, there is a corresponding point on each surface. The piecewise linear curve obtained by connecting corresponding points is given by the dashed curve in Fig. 1. From the figure, it can be observed that the tangent directions determined by the piecewise linear curve are piecewise constants. As $\vec{t}$ is varied, the field of tangent directions obtain their smoothness (level of differentiability) only in $\vec{t}$. To obtain smoothness in going from bounding surface to bounding surface, a sufficiently smooth interpolation must be performed. The result is a smooth vector field of undetermined magnitude which gives the desired tangential directions for coordinate curves connecting the bounding surfaces. A unique vector field of tangents is then obtained by correctly choosing magnitudes so that, on integration, the bounding surfaces are fit precisely.

In symbols, a vector field tangent to the piecewise linear curves is given by

$$\vec{V}_k(\vec{t}) = A_k[\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \ , \qquad (1)$$

between the kth and $(k + 1)$th surfaces where k is taken to vary (if $N > 2$) from the first bounding surface to the final intermediate surface. These vectors are indicated in Fig. 1. The coefficients A_k are scalars which determine the magnitude of the vectors but not the directions. When an independent variable r is assumed for the spanning direction, a partition $r_1 < \ldots < r_{N-1}$ can be specified in correspondence with the tangents of Eq. 1. The partitioned variable can then be used to represent the tangents as the discrete vector valued function which maps r_k into $\vec{V}_k$ for $k = 1 \ldots N-1$. A sufficiently smooth vector field $\vec{V}(r,\vec{t})$ is then obtained by a sufficiently smooth interpolation $\vec{V}(r_k,\vec{t}) = \vec{V}_k(\vec{t})$. With r as a continuous independent variable, the r-derivative of the coordinate transformation $\vec{P}(r,\vec{t})$ is equal to the interpolant and is given by

$$\frac{\partial \vec{P}}{\partial r} = \vec{V} = \sum_{k=1}^{N-1} \psi_k(r)\vec{V}_k(\vec{t}) \ , \tag{2}$$

where $\psi_k(r_j)$ is unity at $k = j$ and vanishes otherwise. When Eq. 2 is integrated with an initial r_1 value of $\vec{P}_1(\vec{t})$, the transformation becomes

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + \sum_{k=1}^{N-1} A_k G_k(r)[\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \ , \tag{3}$$

where

$$G_k(r) = \int_{r_1}^{r} \psi_k(x)dx \ , \tag{4}$$

from which we observe that the interpolants ψ_k must be continuously differentiable up to an order which is one less than the level of smoothness desired for the coordinates. The construction of local controls on the coordinates will rely upon the development of suitably smooth interpolation functions. If the magnitudes A_k of Eq. 1 are chosen so that each $A_k G_k(r_{N-1})$ is unity, then the evaluation of the transformation at r_{N-1} will reduce to $\vec{P}_N(t)$ by means of a telescopic collapse of terms in Eq. 3. With this choice, we obtain the general multisurface transformation of Eiseman [1] which is given by

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + \sum_{k=1}^{N-1} \frac{G_k(r)}{G_k(r_{N-1})} [\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \ . \tag{5}$$

On examination, each interpolation function ψ_k can be rescaled without changing the transformation; hence, the original vector field interpolation becomes an interpolation only on vector directions.

When the interpolants ψ_k are polynomials in r, the coordinate curves which connect the bounding surfaces are globally defined by polynomials in r of one greater degree. The specification of boundary properties for the curves and a global control over their general form are obtained by choices of intermediate surfaces and the associated partitions of r. For notational simplicity, let $r_1 = 0$ and $r_{N-1} = 1$. In the simplest case when there are no intermediate control surfaces, there is just one vector field direction $\vec{V}_1(\vec{t})$ which is determined solely by the bounding surfaces. The interpolant ψ_1 is then a constant function, $G_1(r) = r\psi_1$, $G_1(r)/G_1(1) = r$, and the polynomial 2-surface transformation becomes

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + r[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})] \ . \tag{6}$$

which is the case of linear coordinate curves connecting boundaries. The linear case has occurred in many studies including [2], [3], and [4] and is limited to at most one prescribed coordinate property per boundary which can be either a pointwise distribution or a distribution of angles with the linear transverse coordinate curves. To allow for the prescription of an additional coordinate property on one of the boundaries, an intermediate control surface is introduced and the polynomial 3-surface transformation is computed from Eq. 3 with $\psi_1 = 1 - r$ and $\psi_2 = r$ corresponding to directions of $\vec{V}_1(\vec{t})$ and $\vec{V}_2(\vec{t})$ of Eq. 1 and Fig. 1. The integrals from Eq. 4 become

$$G_1(r) = r - \frac{r^2}{2} ,$$
$$G_2(r) = \frac{r^2}{2} , \tag{7}$$

and since $G_1(1) = G_2(1) = \frac{1}{2}$, the original vector field which is discrete in r is determined by

$$\vec{V}_1(\vec{t}) = 2[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})],$$
$$\vec{V}_2(\vec{t}) = 2[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})], \tag{8}$$

because $A_k = 1./G_k(1)$ in Eq. 1. Upon substitution from Eq. 7 the polynomial 3-surface transformation is given by

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + r(2 - r)[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})]$$
$$+ r^2[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})]. \tag{9}$$

In continuation, an additional coordinate property can be prescribed on each boundary when two intermediate control surfaces are used. The polynomial 4-surface transformation is computed from Eq. 3 with interpolants

$$\psi_1(r) = (1 - r)(r_2 - r),$$

$$\psi_2(r) = r(1 - r), \tag{10}$$

$$\psi_3(r) = (r - r_2)r,$$

which respectively correspond to the directions of $\vec{V}_1(\vec{t})$, $\vec{V}_2(\vec{t})$, and $\vec{V}_3(\vec{t})$ which, in turn, are respectively associated with partition points $r_1 = 0$, r_2, and $r_3 = 1$ and which are defined to vanish at all partition points except the ones of association for each function. For simplicity, we will set $r_2 = \frac{1}{2}$ so that the partition is uniform. The nonuniform case is a simple but algebraically more complex extension [1]. With $r_2 = \frac{1}{2}$, the integrals from Eq. 4 become

$$G_1(r) = \frac{1}{2} r - \frac{3}{4} r^2 + \frac{1}{3} r^3,$$

$$G_2(r) = \frac{1}{2} r^2 - \frac{1}{3} r^3, \tag{11}$$

$$G_3(r) = \frac{1}{3} r^3 - \frac{1}{4} r^2,$$

and from an evaluation at the endpoint $r = 1$ we have $G_1(1) = \frac{1}{12}$, $G_2(1) = \frac{1}{6}$, and $G_3(1) = \frac{1}{12}$. By substitution, the polynomial 4-surface transformation is given by

$$\vec{\beta}(r,\vec{t}) = \vec{\beta}_1(\vec{t}) + r(6 - 9r + 4r^2)[\vec{\beta}_2(\vec{t}) - \vec{\beta}_1(\vec{t})]$$

$$+ r^2(3 - 2r)[\vec{\beta}_3(\vec{t}) - \vec{\beta}_2(\vec{t})] \qquad (12)$$

$$+ r^2(4r - 3)[\vec{\beta}_4(\vec{t}) - \vec{\beta}_3(\vec{t})],$$

with r-derivative

$$\frac{\partial\vec{\beta}}{\partial r} = 6(1 - r)(1 - 2r)[\vec{\beta}_2(\vec{t}) - \vec{\beta}_1(\vec{t})]$$

$$+ 6r(1 - r)[\vec{\beta}_3(\vec{t}) - \vec{\beta}_2(\vec{t})] \qquad (13)$$

$$+ 6r(2r - 1)[\vec{\beta}_4(\vec{t}) - \vec{\beta}_3(\vec{t})] .$$

By direct evaluation

$$\vec{\beta}(0,\vec{t}) = \vec{\beta}_1(\vec{t}),$$

$$\qquad (14)$$

$$\vec{\beta}(1,\vec{t}) = \vec{\beta}_4(\vec{t}),$$

and

$$\frac{\partial\vec{\beta}}{\partial r}(0,\vec{t}) = 6[\vec{\beta}_2(\vec{t}) - \vec{\beta}_1(\vec{t})],$$

$$\qquad (15)$$

$$\frac{\partial\vec{\beta}}{\partial r}(1,\vec{t}) = 6[\vec{\beta}_4(\vec{t}) - \vec{\beta}_3(\vec{t})],$$

which explicitly shows that in addition to fitting the boundaries (Eq. 14), the intermediate surfaces $\vec{P}_2(t)$ and $\vec{P}_3(\vec{t})$ can be used to control the angles at which the transverse coordinate curves in r intersect the boundaries (Eq. 15). Moreover, the choice of intermediate surfaces can also be used to control the shape of the transverse curves and the distribution of the constant r coordinate surfaces. The general derivation and discussion is given in Eiseman [1]. For our purposes, the discussion on coordinate system controls will be deferred until local methods are presented for our survey of some of the material developed in Eiseman [5], [6].

An alternative form of the polynomial 4-surface transformation (Eq. 12) can be obtained from the evaluations of the transformation (Eq. 14) and its derivatives (Eq. 15). With the evaluations, the intermediate surfaces can be expressed entirely in terms of boundary data, which results in

$$\vec{P}_2(\vec{t}) = \vec{P}(0,\vec{t}) + \frac{1}{6}\frac{\partial\vec{P}}{\partial r}(0,\vec{t}),$$

(16)

and

$$\vec{P}_3(\vec{t}) = \vec{P}(1,\vec{t}) - \frac{1}{6}\frac{\partial\vec{P}}{\partial r}(1,\vec{t}).$$

Upon substitution of Eqs. 14 and 16 into Eq. 12 we obtain

$$\begin{aligned}\vec{P}(r,\vec{t}) = {} & (1 - 3r^2 + 2r^3)\vec{P}(0,\vec{t}) + r^2(3 - 2r)\vec{P}(1,\vec{t}) \\ & + r(1 - r)^2 \frac{\partial\vec{P}}{\partial r}(0,\vec{t}) - r^2(1 - r)\frac{\partial\vec{P}}{\partial r}(1,\vec{t}).\end{aligned} \tag{17}$$

after grouping terms by boundary type. By examination, the coefficients of the boundary evaluations for the transformation and its r-derivative can easily be identified as just the canonical Hermite cubic interpolants on the unit interval $0 \leq r \leq 1$. When the r-derivatives are specified to be normal to the respective boundaries, we obtain the transformation presented by Smith and Weigel [3].

In continuation, polynomial N-surface transformations can be systematically established from Eqs. 3-4 and the interpolants

$$\psi_k(r) = \prod_{\substack{i=1 \\ i \neq k}}^{N-1} (r - r_i), \tag{18}$$

for $k = 1,2,\ldots,N-1$. In each case, the transverse coordinate curves are polynomials of degree N-1 in r with vector valued coefficients which are functions of the surface coordinates $\vec{t}$. Polynomials, however, are globally defined for all r, and as a consequence, local mesh properties cannot be controlled without a global effect. As an example, suppose that we wish to smoothly embed a general rectilinear Cartesian system within a global mesh structure to obtain a system of the form illustrated in Fig. 2 where the Cartesian region within the mesh is bounded by the darkened curves. In the Cartesian part of the mesh, coordinate curves in r would be lines which pass through it at a uniform rate. Since global polynomials in r would be uniquely determined by the Cartesian region, curved boundaries could not be fitted.

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL
TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS
FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED
BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B10
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR
FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR
FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO
REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY
COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX
FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN
SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING
UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

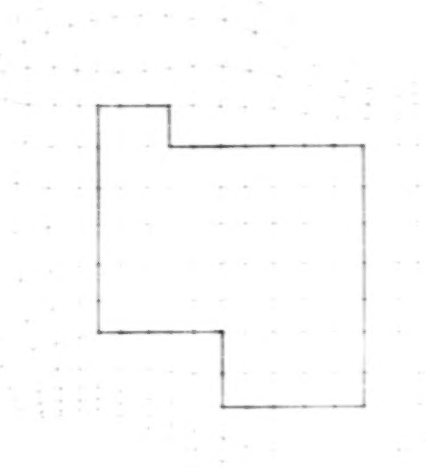


Fig. 2 - A smoothly embedded Cartesian region
within a global mesh structure.

To obtain precise local controls which could be successfully applied to generate a mesh as illustrated above in Fig. 2, local forms of the multisurface transformation (Eqs. 3-4) were established and analyzed by Eiseman [5], [6], and [7]. Our discussion will follow the development given by Eiseman in [6] and will focus upon two-dimensional applications with a surface coordinate $\vec{t} = t$. When the interpolants ψ_k are nonvanishing on only a local region, the precise local controls over the coordinates that were obtained will be illustrated with the local piecewise linear interpolants that are depicted in Fig. 3. For algebraic simplicity, the analysis is restricted to the case with the uniform partition $r_k = k$ with the clear understanding that nonuniform partitions will follow the same analytic pattern. Since the multi-surface transformation remains unchanged when the interpolants are scaled by any sequence of nonzero

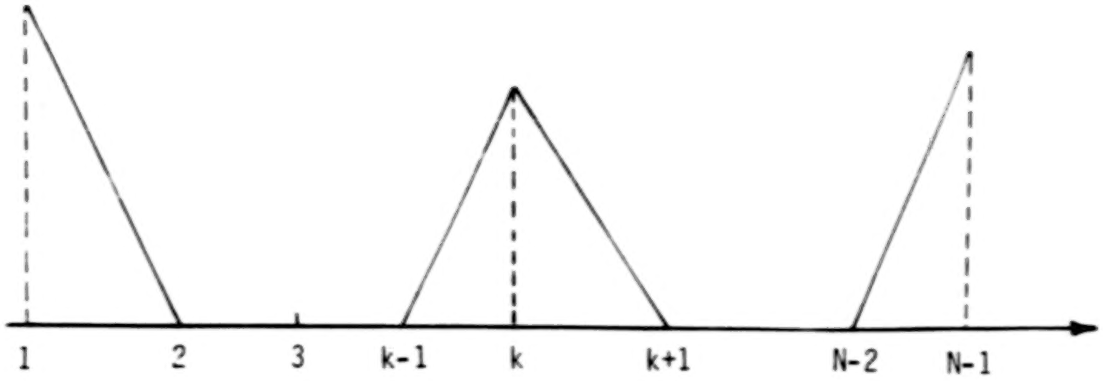


Fig. 3 - Piecewise linear local interpolants with partition points $r_k = k$ for $k = 1,2,\ldots,N-1$.

numbers, the height $\psi_k(r_k)$ of each interpolant can be chosen arbitrarily. In particular, the form of the multi-surface transformation can be simplified when the heights are adjusted so that each interpolant integrates to unity which yields $G_k(r_{N-1}) = 1$ for all k. The integrals are obtained from triangular areas, and by direct observation, lead to the height adjustments $\psi_1(r_1) = 2$, $\psi_k(r_k) = 1$, and $\psi_{N-1}(r_{N-1}) = 2$ in correspondence with the successive illustrations in Fig. 3. Also, in correspondence, the explicit form of the normalized interpolants are given by

$$\psi_1(r) = \left\{\begin{array}{ll} 2(2 - r) & \text{for } 1 \le r < 2 \\ & \\ 0 & \text{for } 2 \le r \le N - 1 \end{array}\right\}; \qquad (19)$$

$$\psi_k(r) = \left\{ \begin{array}{ll} 0 & \text{for } 1 \leq r < k - 1 \\ (r - k) + 1 & \text{for } k - 1 \leq r < k \\ (k - r) + 1 & \text{for } k \leq r < k + 1 \\ 0 & \text{for } k + 1 \leq r \leq N - 1 \end{array} \right\}; \qquad (20)$$

$$\psi_{N-1}(r) = \left\{ \begin{array}{ll} 0 & \text{for } 1 \leq r < N - 2 \\ 2(r - N + 2) & \text{for } N - 2 \leq r \leq N - 1 \end{array} \right\}; \qquad (21)$$

and their integrals defined in Eq. 4, by

$$G_1(r) = \left\{ \begin{array}{ll} 1 - (2 - r)^2 & \text{for } 1 \leq r < 2 \\ 1 & \text{for } 2 \leq r \leq N - 1 \end{array} \right\}; \qquad (22)$$

$$G_k(r) = \left\{ \begin{array}{ll} 0 & \text{for } 1 \leq r < k - 1 \\ 1/2(r - k)^2 + (r - k) + 1/2 & \text{for } k - 1 \leq r < k \\ -1/2(k - r)^2 - (k - r) + 1/2 & \text{for } k \leq r < k + 1 \\ 1 & \text{for } k + 1 \leq r \leq N - 1 \end{array} \right\}; \qquad (23)$$

$$G_{N-1}(r) = \left\{ \begin{array}{ll} 0 & \text{for } 1 \leq r < N - 2 \\ (r - N + 2)^2 & \text{for } N - 2 \leq r \leq N - 1 \end{array} \right\}, \qquad (24)$$

which are depicted in Fig. 4.

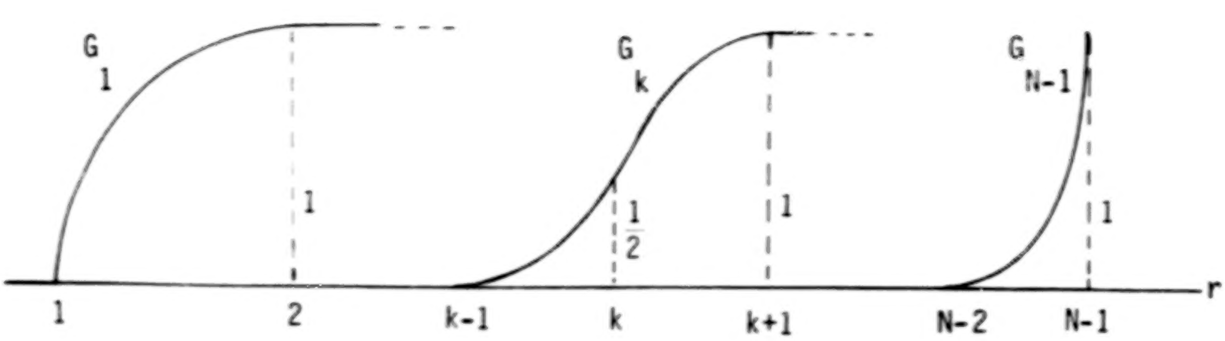


Fig. 4 - Integrals of normalized interpolants for the partition $r_k = k$.

On the interval $k \leq r < k + 1$, the integrals $G_i(r)$ which correspond to interpolants defined over nonintersecting intervals are either unity or vanishing depending upon whether the interval of definition precedes or follows the interval under examination. When $i = 1,2,\ldots,k - 1$ which is nonvacuous for $k > 1$, the integrals $G_i(r)$ have been evaluated over the entire domain for which the respective interpolant ψ_i is nonvanishing; hence, these preceding integrals are unity by the chosen normalization. When $i = k + 2, k + 3,\ldots,N - 1$, the interpolants ψ_i each vanish on $1 \leq r < k + 1$; hence, the integrals G_i also vanish there. As a consequence, G_i for $i = k, k + 1$ yield the only nontrivial contributions for the multi-surface transformation which reduces to

$$\vec{P}(r,t) = \vec{P}_1(t) + \sum_{i=1}^{k-1} [\vec{P}_{i+1}(t) - \vec{P}_i(t)] + G_k(r)[\vec{P}_{k+1}(t) - \vec{P}_k(t)]$$

$$+ G_{k+1}(r)[\vec{P}_{k+2}(t) - \vec{P}_{k+1}(t)] + 0 \qquad (25)$$

$$= \vec{P}_k(t) + G_k(r)[\vec{P}_{k+1}(t) - \vec{P}_k(t)] + G_{k+1}(r)[\vec{P}_{k+2}(t) - \vec{P}_{k+1}(t)],$$

which depends upon only the three control surfaces $\vec{P}_k$, $\vec{P}_{k+1}$, $\vec{P}_{k+2}$ which can be arbitrarily selected to our advantage when they are not bounding surfaces. With substitutions from Eqs. 22-24, we obtain the partition point ($r_k = k$ for $k = 1,2,\ldots,N-1$) evaluations

$$\begin{aligned}
\vec{P}(1,t) &= \vec{P}_1(t), \\
\vec{P}(2,t) &= \tfrac{1}{2}\,[\vec{P}_2(t) + \vec{P}_3(t)], \\
&\ \ \vdots \\
\vec{P}(k,t) &= \tfrac{1}{2}\,[\vec{P}_k(t) + \vec{P}_{k+1}(t)], \\
\vec{P}(k+1,t) &= \tfrac{1}{2}\,[\vec{P}_{k+1}(t) + \vec{P}_{k+2}(t)], \\
&\ \ \vdots \\
\vec{P}(N-2,t) &= \tfrac{1}{2}\,[\vec{P}_{N-2}(t) + \vec{P}_{N-1}(t)], \\
\vec{P}(N-1,t) &= \vec{P}_N(t),
\end{aligned} \qquad (26)$$

which, in addition to boundary fitting at the end points $r = 1$ and $r = N - 1$, also shows that the transformation passes through the midpoints of the lines which connect the intermediate control surfaces for any fixed surface coordinate t. Moreover, from the general multi-surface construction (Eqs. 1-5 and Fig. 1), the transverse coordinate curves are tangent to the connecting lines at the partition point evaluations. The tangents at partition points can be explicitly obtained from substitutions of the interpolation functions (Eqs. 19-21) into the r-derivative of the transformation (Eq. 25) and are given by

$$
\begin{aligned}
\frac{\partial \vec{P}}{\partial r}(1,t) &= 2[\vec{P}_2(t) - \vec{P}_1(t)], \\
\frac{\partial \vec{P}}{\partial r}(2,t) &= \vec{P}_3(t) - \vec{P}_2(t), \\
&\vdots \\
\frac{\partial \vec{P}}{\partial r}(k,t) &= \vec{P}_{k+1}(t) - \vec{P}_k(t), \\
\frac{\partial \vec{P}}{\partial r}(k+1,t) &= \vec{P}_{k+2}(t) - \vec{P}_{k+1}(t), \\
&\vdots \\
\frac{\partial \vec{P}}{\partial r}(N-2,t) &= \vec{P}_{N-1}(t) - \vec{P}_{N-2}(t), \\
\frac{\partial \vec{P}}{\partial r}(N-1,t) &= 2[\vec{P}_N(t) - \vec{P}_{N-1}(t)].
\end{aligned}
\tag{27}
$$

In graphical form, this process is depicted in Fig. 5.

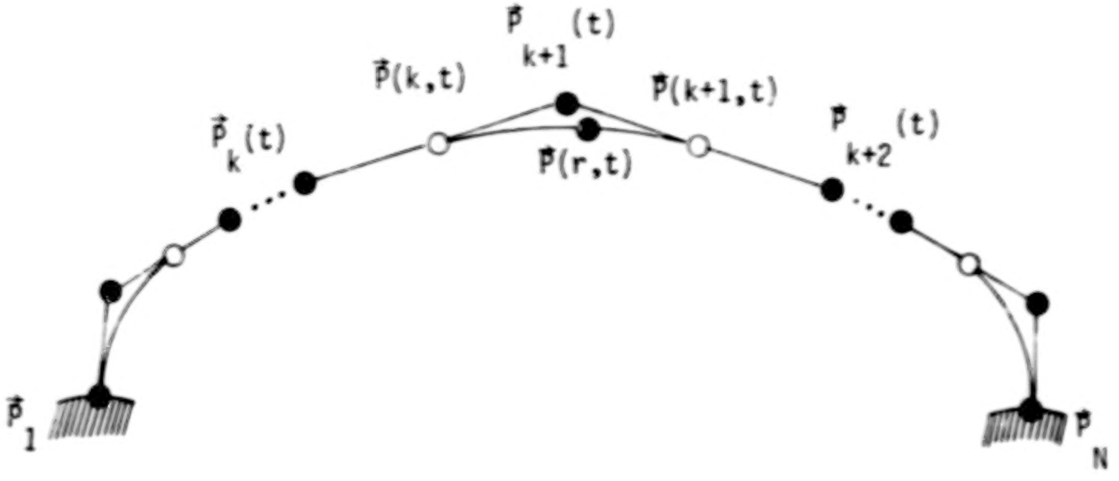

$\vec{P}_{k+1}(t)$
$\vec{P}(k,t)$
$\vec{P}(k+1,t)$
$\vec{P}_k(t)$
$\vec{P}(r,t)$
$\vec{P}_{k+2}(t)$
$\vec{P}_1$
$\vec{P}_N$

Fig. 5 - Coordinate curve segments for $k \le r < k+1$.

Between the control surfaces $\vec{P}_i$ and $\vec{P}_j$ for $i > j$, the distribution of constant r coordinate surfaces can be controlled for the general multisurface transformation (Eqs. 4-5) when uniformity can be specified along a direction of measurement

$$\hat{\imath}(\vec{t}) = \frac{\vec{P}_i(\vec{t}) - \vec{P}_j(\vec{t})}{||\vec{P}_i(\vec{t}) - \vec{P}_j(\vec{t})||} , \tag{28}$$

for then arbitrary distributions can be applied relative to uniform conditions. An illustration is given in Fig. 6.

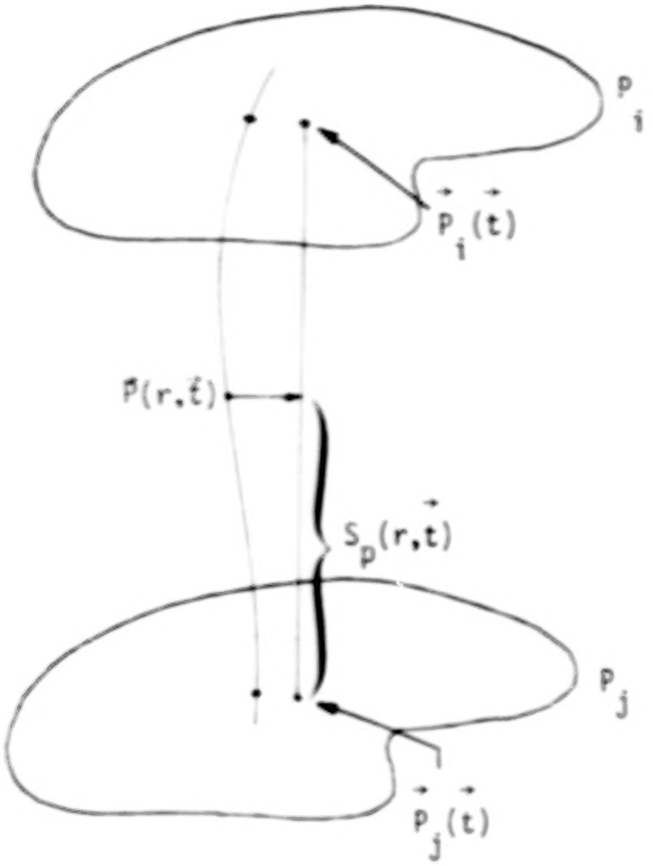

P_i
$\vec{P}_i(\vec{t})$
$\vec{P}(r,\vec{t})$
$S_p(r,\vec{t})$
P_j
$\vec{P}_j(\vec{t})$

Fig. 6 - Measurement of uniformity.

To obtain uniformity, the projected arc length

$$S_p(r,\vec{t}) = [\vec{P}(r,\vec{t}) - \vec{P}_j(\vec{t})] \cdot \hat{\tau}(\vec{t}), \tag{29}$$

depicted in Fig. 6 must be linear in r, or equivalently $\partial S_p/\partial r$ must be independent of r. But then from Eq. 25 and with the relative projections $C_m(t) = [\vec{P}_{m+1}(t) - \vec{P}_m(t)] \cdot \hat{\tau}(t)$, we have

$$\frac{\partial S_p}{\partial r} = \phi_k(r)C_k(t) + \phi_{k+1}(r)C_{k+1}(t)$$

$$= \left\{ \begin{array}{ll} -2C_1(t) + C_2(t) & \text{for } k = 1 \\ -C_k(t) + C_{k+1}(t) & \text{for } 1 < k < N - 1 \\ -C_{N-2}(t) + 2C_{N-1}(t) & \text{for } k = N - 1 \end{array} \right\} r + \text{function of } t, \quad (30)$$

where the last equality comes from Eqs. 19-21. Hence, for $k = j, j+1, \ldots, i-1$, uniformity is obtained if $2C_1(t) = C_2(t)$ should $k = 1$, $C_k(t) = C_{k+1}(t)$ should $1 < k < N - 1$, and $C_{N-2}(t) = 2C_{N-1}(t)$ should $k = N - 1$. A more thorough discussion on uniformity is available in Eiseman [1] for the global case, in Eiseman [5], [6] for the local case, and in Eiseman [7] for the general cases.

To explicitly demonstrate the application of the local controls, and at the same time, reveal the basic algorithmic steps, coordinates will be obtained for a simple transition from a purely Cartesian system into a purely Polar system. For $0 \leq t \leq 1$, the Cartesian coordinates will be specified below a line $\hat{O}(t) = (2t-1,0)$ and the Polar coordinates, beyond a circular arc $\sqrt{2}\,\hat{u}(t)$ where $\hat{u}(t) = (-\cos\theta, \sin\theta)$ for $\theta = (2t + 1)\pi/4$. The line and the arc are depicted in Fig. 7.

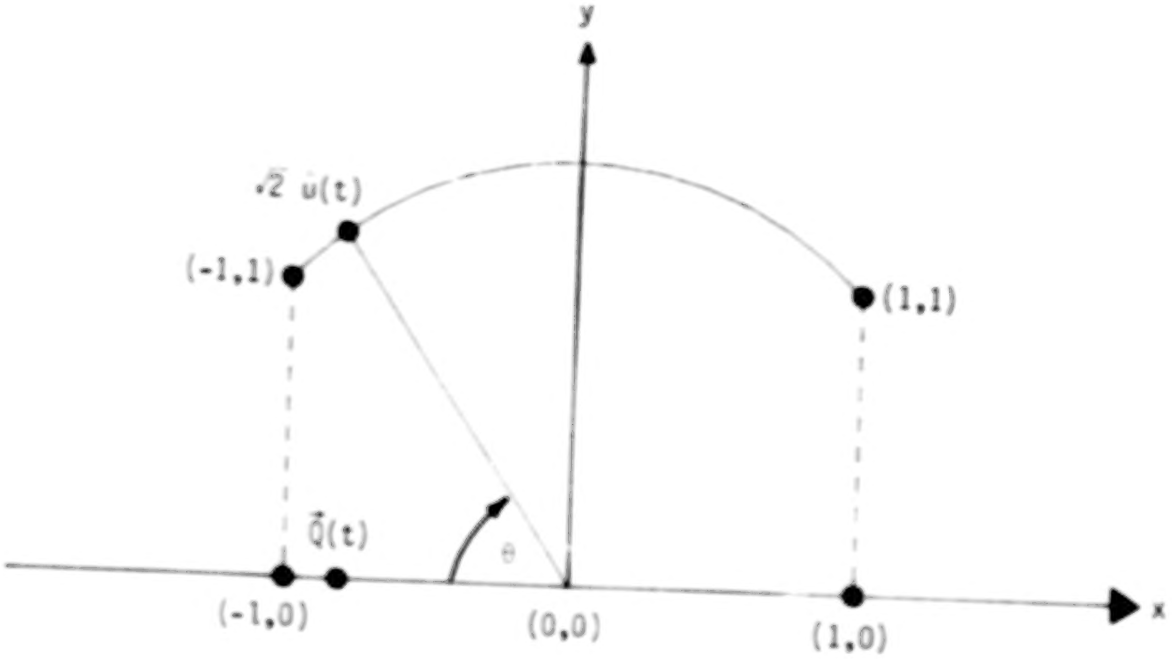


Fig. 7 - Basic curves for the Cartesian to Polar transition example.

To obtain uniformity near the sides ($t = 0,1$) of the transitional region, the unit vertical distance will be used as a basis for displacements to establish uniformity below the line and beyond the circular arc. For the line,

$$\begin{aligned}
\vec{P}_1(t) &= \vec{Q}(t) - (0,\tfrac{5}{2}),\\
\vec{P}_2(t) &= \vec{Q}(t) - (0,2),\\
\vec{P}_3(t) &= \vec{Q}(t) - (0,1), \qquad (31)\\
\vec{P}_4(t) &= \vec{Q}(t).
\end{aligned}$$

so that for

$$\hat{\imath}(t) = \frac{\vec{P}_4(t) - \vec{P}_1(t)}{||\vec{P}_4(t) - \vec{P}_1(t)||} = (0,1), \tag{32}$$

we have

$$\begin{aligned} C_1(t) &= [\vec{P}_2(t) - \vec{P}_1(t)] \cdot \hat{\imath}(t) = (0,\tfrac{1}{2}) \cdot (0,1) = \tfrac{1}{2}, \\ C_2(t) &= [\vec{P}_3(t) - \vec{P}_2(t)] \cdot \hat{\imath}(t) = (0,1) \cdot (0,1) = 1, \\ C_3(t) &= [\vec{P}_4(t) - \vec{P}_3(t)] \cdot \hat{\imath}(t) = (0,1) \cdot (0,1) = 1, \end{aligned} \tag{33}$$

which satisfies uniformity for $1 \leq r \leq 3$ and yields a Cartesian system from

$$\vec{P}(1,t) = \vec{P}_1(t) = \vec{Q}(t) - (0,\tfrac{5}{2}), \tag{34}$$

up to

$$\vec{P}(3,t) = \frac{1}{2}\,[\vec{P}_3(t) + \vec{P}_4(t)] = \vec{Q}(t) - (0,\tfrac{1}{2}). \tag{35}$$

Similarly, for the circular arc, let

$$\begin{aligned} \vec{P}_5(t) &= \sqrt{2}\,\hat{u}(t), \\ \vec{P}_6(t) &= (1 + \sqrt{2})\,\hat{u}(t), \end{aligned} \tag{36}$$

$$\vec{P}_7(t) = (\tfrac{3}{2} + \sqrt{2})\, \hat{u}(t),$$

be the last three surfaces so that for

$$\hat{\tau}(t) = \frac{\vec{P}_7(t) - \vec{P}_5(t)}{||\vec{P}_7(t) - \vec{P}_5(t)||} = \hat{u}(t) \tag{37}$$

we have

$$C_5(t) = [\vec{P}_6(t) - \vec{P}_5(t)] \cdot \hat{\tau}(t) = \hat{u}(t) \cdot \hat{u}(t) = 1,$$

and (38)

$$C_6(t) = [\vec{P}_7(t) - \vec{P}_6(t)] \cdot \hat{\tau}(t) = \frac{1}{2}\, \hat{u}(t) \cdot \hat{u}(t) = \frac{1}{2},$$

which satisfies uniformity for $5 \le r \le 6$ and yields a Polar system from the circular arc

$$\vec{P}(5,t) = \frac{1}{2}\, [\vec{P}_5(t) + \vec{P}_6(t)] = (\tfrac{1}{2} + \sqrt{2})\, \hat{u}(t), \tag{39}$$

up to the circular arc

$$\vec{P}(6,t) = \vec{P}_7(t) = (\tfrac{3}{2} + \sqrt{2})\, \hat{u}(t). \tag{40}$$

The entire collection of bounding and intermediate surfaces are depicted in Fig. 8.

Table 1

MESH INDEX	r	k	G_k	G_{k+1}	MESH INDEX	r	k	G_k	G_{k+1}
1	1.00	1	.00	.00	11	3.50	3	.88	.13
2	1.25	1	.44	.03	12	3.75	3	.97	.28
3	1.50	1	.75	.13	13	4.00	4	.50	.00
4	1.75	1	.94	.28	14	4.25	4	.72	.03
5	2.00	2	.50	.00	15	4.50	4	.88	.13
6	2.25	2	.72	.03	16	4.75	4	.97	.28
7	2.50	2	.88	.13	17	5.00	5	.50	.00
8	2.75	2	.97	.28	18	5.25	5	.72	.06
9	3.00	3	.50	.00	19	5.50	5	.88	.25
10	3.25	3	.72	.03	20	5.75	5	.97	.56
					21	6.00	5	1.00	1.00

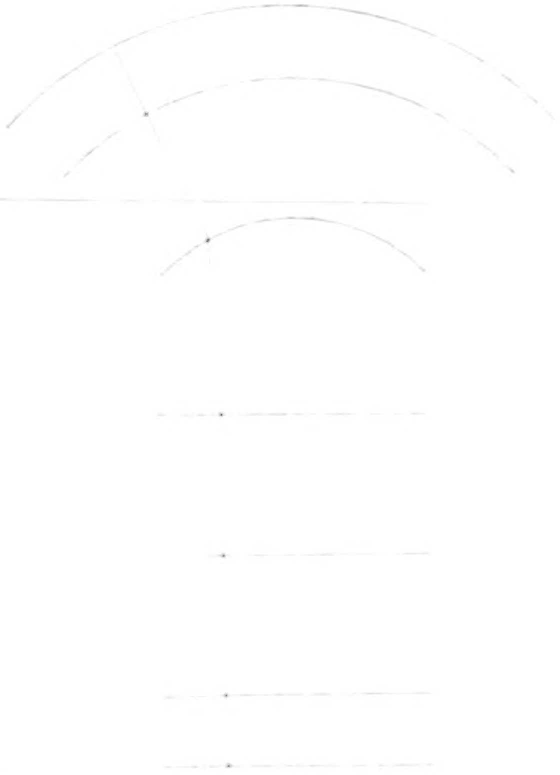


Fig. 8 - Control surfaces for polar-rectangular mesh.

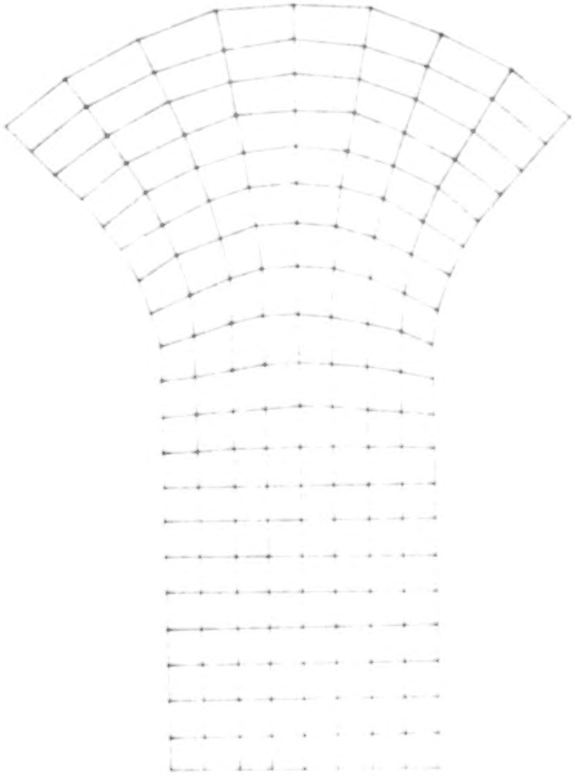


Fig. 9 - Polar-rectangular mesh.

For 21 equally spaced mesh points in r, the evaluation of the r-dependent functions is given with two decimal places of accuracy in the table. For a given mesh point, the interval $k \leq r < k + 1$ containing it determines the index k for G_k and G_{k+1} respectively in Eqs. 22-24. Due to the uniform selection of partition points r_k, a repetitive pattern in the G_k evaluations can be observed and is indicative of translated versions of the same function. When 9 uniformly spaced mesh points are chosen for $0 \leq t \leq 1$, and when the multi-surface transformation of Eq. 9 is evaluated for the 21 × 9 mesh, we obtain the coordinate mesh which is displayed in Fig. 9. From uniformity and Table 1, the first 8 and the last 5 mesh points in r are seen to be uniformly distributed, and the mesh is respectively purely Cartesian and purely Polar for those points. To illustrate the computational aspect, we shall explicitly evaluate the transformation at the point with curvilinear variables $r = 4.5$, $t = 0$. At $t = 0$, we have

$$\vec{Q}(0) = (2(0)-1,0) = (-1,0),$$

and (41)

$$\hat{u}(0) = (-\cos\frac{\pi}{4}, \sin\frac{\pi}{4}) = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}).$$

For $r = 4.5$, we are at the 15th mesh index in Table 1 where we read across to note that we are in the 4th interval $(4 \leq 4.5 \leq 5)$ with $G_4(4.5) = .88$ and $G_5(4.5) = .13$. By substitution into the transformation (Eq. 25 for $k = 4$) we obtain

$$\vec{P}(4.5,0) = \vec{P}_4(0) + G_4(4.5)[\vec{P}_5(0) - \vec{P}_4(0)] + G_5(4.5)[\vec{P}_6(0) - \vec{P}_5(0)]$$

$$= \vec{Q}(0) + G_4(4.5)[\sqrt{2}\,\hat{u}(0) - \vec{Q}(0)] + G_5(4.5)\,\hat{u}(0)$$

$$= (-1,0) + .88[\sqrt{2}(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) - (-1,0)] + .13(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$$

$$= (-1,0) + .88(0,1) + (-.09,09) \qquad (42)$$

$$= (-1 + 0 -.09,\ 0 + .88 + .09)$$

$$= (-1.09,\ .97).$$

In continuation with local methods, the case with nonuniform partitions for the piecewise linear functions is given in Eiseman [5]. In addition, local interpolants with a higher level of smoothness (derivative continuity) can be used and are developed in Eiseman [7]. With the local controls over the transverse coordinate curves which connect two bounding surfaces, lateral bounding surfaces can also be approximately fit. A precise fit of the lateral boundaries can be obtained with blending functions which were used by Gordon and Hall [8] to create a global method. Further applications of blending functions will be presented at this workshop by Ericksson [9], by Forsey, Edwards, and Carr [10] and by Anderson and Spradley [11].

Algebraic Mesh Generation - Three Dimensions

An algebraic approach to mesh generation in three dimensions results in algebraic functions that relate a computational domain to a physical domain. If the computational domain is defined by the three variables r, ξ, and ζ on the unit cube

$$0 \leq r \leq 1 \; ,$$

$$0 \leq \xi \leq 1 \; , \tag{43}$$

$$0 \leq \zeta \leq 1,$$

then the physical domain in Cartesian (x,y,z) coordinates is given by the transformation $\vec{P}(r,\xi,\zeta) = (x,y,z)$ where

$$x = x(r,\xi,\zeta) \; ,$$

$$y = y(r,\xi,\zeta) \; , \tag{44}$$

$$z = z(r,\xi,\zeta).$$

When Eq. 44 is nonsingular it has an inverse transformation denoted by

$$
\begin{aligned}
r &= r(x,y,z) , \\
\xi &= \xi(x,y,z) , \\
\zeta &= \zeta(x,y,z) .
\end{aligned}
\tag{45}
$$

A uniform mesh is defined on the computational domain by constants Δr, $\Delta\xi$, $\Delta\zeta$ (Fig. 10). This mesh maps using Eq. 44 to a corresponding mesh

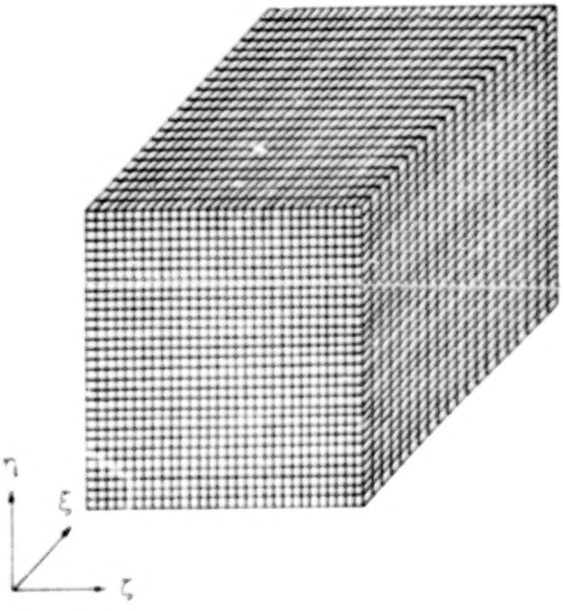


Fig. 10 - Computational domain.

in the physical domain which is not necessarily uniform. A simple example for Eq. 44 is given by

$$x = \left(\frac{e^{k_1\xi} - 1}{e^{k_1} - 1}\right) X_L,$$

$$y = \xi X_L\{\tan\theta_y + Y_L\left(\frac{e^{k_2 r} - 1}{e^{k_2} - 1}\right)\}, \tag{46}$$

$$z = \xi X_L\{\tan\theta_z + Z_L\left(\frac{e^{k_3\zeta} - 1}{e^{k_3} - 1}\right)\},$$

or

$$\xi = \frac{1}{k_1}\ln\{1 + (e^{k_1} - 1)\frac{x}{X_L}\},$$

$$r = \frac{1}{k_2}\ln\{1 + (e^{k_2} - 1)(\frac{y}{x} - \tan\theta_y)\frac{1}{Y_L}\}, \tag{47}$$

$$\zeta = \frac{1}{k_3}\ln\{1 + (e^{k_3} - 1)(\frac{z}{x} - \tan\theta_z)\frac{1}{Z_L}\},$$

where k_1, k_2, k_3, θ_y, θ_z, X_L, Y_L, and Z_L are constants. For $\xi_0 \leq \xi \leq 1$, $0 \leq r \leq 1$, $0 \leq \zeta \leq 1$ and $Y_L = Z_L$ the uniform computational domain maps into a frustrum of a paramid (Fig. 11) and the mesh is concentrated in the physical domain according to the magnitudes and signs of k_1, k_2, and k_3.

Equation 44 must satisfy the constraints outlined in the introduction and which vary from problem to problem. For many mesh generation problems, the constraints reduce to having the boundaries in the computational domain map to boundaries in the physical

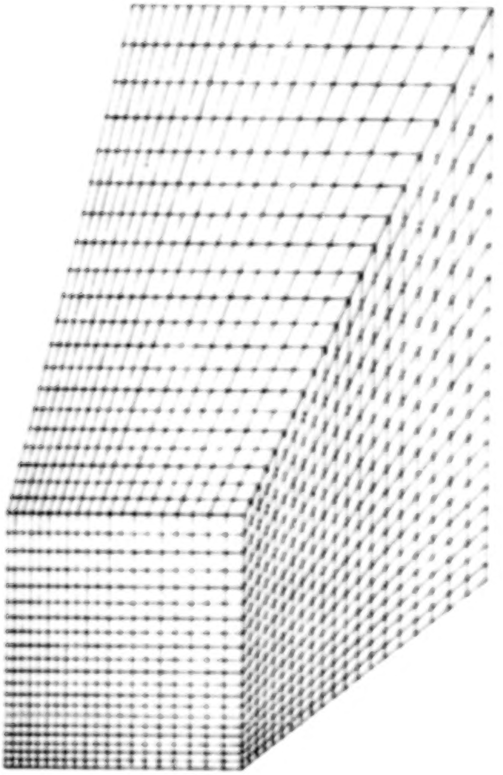


Fig. 11 - Physical domain for Eqs. 46-47.

domain and concentrating the mesh in specified regions of the physical domain. The polynomial N-surface transformations (Eq. 6-18) are global algebraic mesh generation techniques which satisfy the basic boundary constraints and result in polynomial functions of degree $(N - 1)$ with respect to one of the independent variables. For a small N the polynomials are particularly simple. If the surface coordinates are $\vec{t} = (\xi,\zeta)$, the transformation $\vec{P}(r,\vec{t}) = (x(r,\vec{t}),\ y(r,\vec{t}),\ z(r,\vec{t}))$

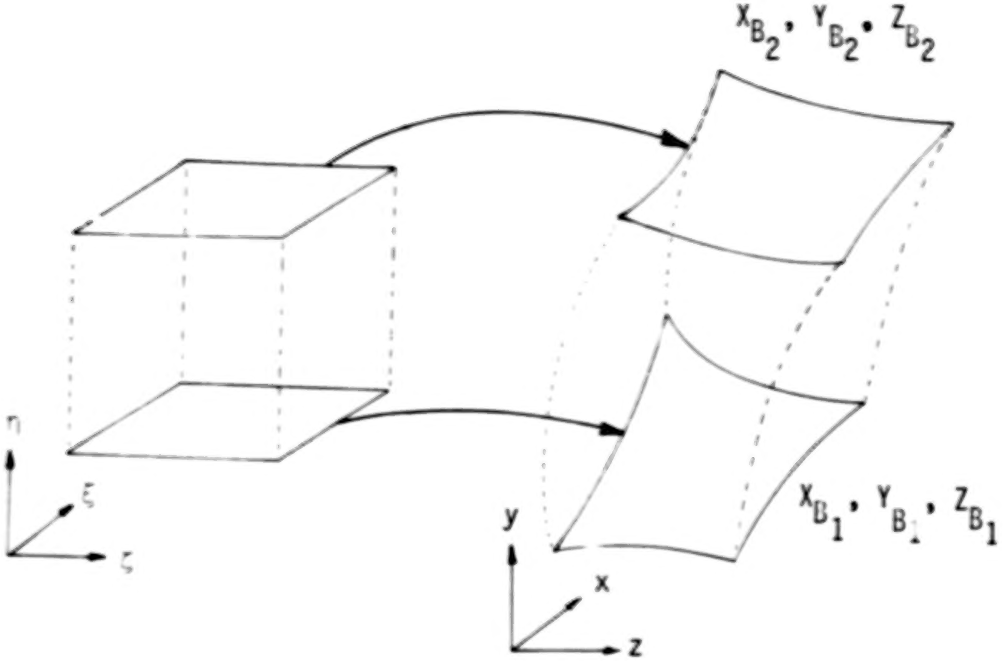

$X_{B_2}, Y_{B_2}, Z_{B_2}$
$X_{B_1}, Y_{B_1}, Z_{B_1}$
η
ξ
ζ
y
x
z

Fig. 12 - Boundary mapping.

is defined such that

$$
\begin{aligned}
X_{B_1} &= x(0,\xi,\zeta) = X_1(\xi,\zeta), \\
Y_{B_1} &= y(0,\xi,\zeta) = Y_1(\xi,\zeta), \\
Z_{B_1} &= z(0,\xi,\zeta) = Z_1(\xi,\zeta), \\
X_{B_2} &= x(1,\xi,\zeta) = X_2(\xi,\zeta), \\
Y_{B_2} &= y(1,\xi,\zeta) = Y_2(\xi,\zeta), \\
Z_{B_2} &= z(1,\xi,\zeta) = Z_2(\xi,\zeta),
\end{aligned}
\qquad (48)
$$

where $\vec{P}_1(\xi,\zeta) = (X_{B_1}, Y_{B_1}, Z_{B_1})$ is one boundary and $\vec{P}_N(\xi,\zeta) = (X_{B_2}, Y_{B_2}, Z_{B_2})$ is the other boundary in the physical domain (Fig. 12).

The polynomial 4-surface transformation (Eq. 12 and 17) allows a constraint to be placed on the mesh in addition to that of fitting the boundaries. This constraint occurs when the physical mesh is required to be orthogonal at the boundaries. Since the derivatives $\frac{\partial X}{\partial r}(0,\xi,\zeta)$, $\frac{\partial X}{\partial r}(1,\xi,\zeta)$, etc. can be computed from the cross product of the tangential derivatives $\frac{dX_1}{d\xi}(\xi,\zeta)$, $\frac{dX_1}{d\zeta}(\xi,\zeta)$, $\frac{dY_1}{d\xi}(\xi,\zeta)$, $\frac{dY_1}{d\zeta}(\xi,\zeta)$, etc., we have

$$\frac{\partial X}{\partial r}(\ell-1,\xi,\zeta)\,\vec{i} + \frac{\partial Y}{\partial r}(\ell-1,\xi,\zeta)\,\vec{j} + \frac{\partial Z}{\partial r}(\ell-1,\xi,\zeta)\,\vec{k} =$$

$$K \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial X_\ell}{\partial \xi}(\xi,\zeta) & \frac{\partial Y_\ell}{\partial \xi}(\xi,\zeta) & \frac{\partial Z_\ell}{\partial \xi}(\xi,\zeta) \\ \frac{\partial X_\ell}{\partial \zeta}(\xi,\zeta) & \frac{\partial Y_\ell}{\partial \zeta}(\xi,\zeta) & \frac{\partial Z_\ell}{\partial \zeta}(\xi,\zeta) \end{vmatrix} \quad \ell = 1,2 \qquad (49)$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors and K is the magnitude of the normal vector, the choice of which can be used to apply controls developed in Eiseman [1] for the shape of coordinate curves in r and for the distribution of constant r-surfaces. Applying this procedure will force the mesh to be orthogonal at the boundaries but not necessarily anywhere else.

A globally uniform computational mesh (for linear S_p in Fig. 6) can be mapped onto a physical mesh with the polynomial N-surface transformations given in Eqs. 6, 9, and 12. Concentration of mesh points in the r direction is accomplished by choosing a function $r(r)$ such that

$0 \leq r \leq 1$, $0 \leq \bar{r} \leq 1$ and $\frac{d\bar{r}}{dr} > 0$ (Fig. 13). For example Smith and

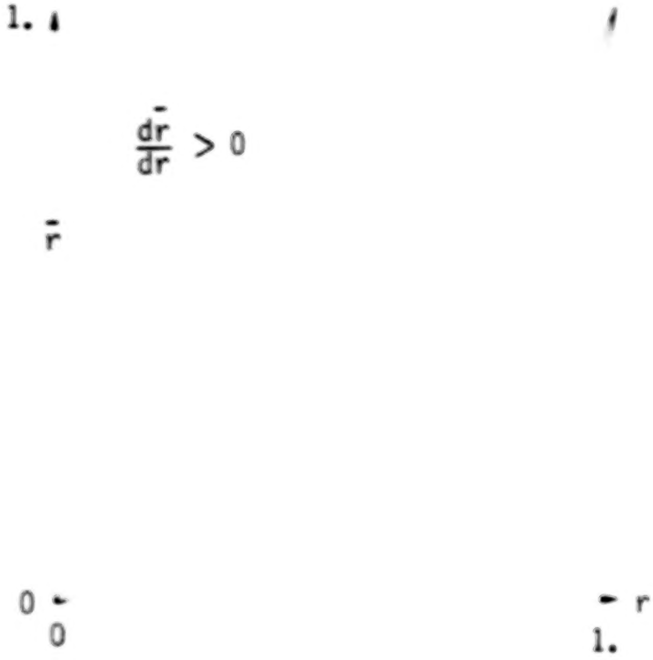


Fig. 13 - Grid control.

Weigel [3] used the function

$$\bar{r} = \frac{e^{kr}-1}{e^{k}-1} \quad ; \quad 0 \leq r \leq 1, \tag{50}$$

to contract the physical grid toward one boundary or the other. The number k is a free parameter whose magnitude dictates the degree of contraction. When r is replaced by $\bar{r}(r)$ in Eq. 6, the contractive function becomes embedded in the linear polynomial part of Eq. 6, which results in

$$\begin{aligned} x &= X_2(\xi,\zeta)\bar{r} + X_1(\xi,\zeta)(1 - \bar{r}), \\ y &= Y_2(\xi,\zeta)\bar{r} + Y_1(\xi,\zeta)(1 - \bar{r}), \\ z &= Z_2(\xi,\zeta)\bar{r} + Z_1(\xi,\zeta)(1 - \bar{r}), \end{aligned} \tag{51}$$

where a uniform partitio[illegible] $r \leq 1$ yields a desired nonuniform partition of $0 \leq \bar{r} \leq 1$ that, in [illegible] [illegible]portionately partitions the linear segments of the transformation.

The example previously presented can be derived with this approach where

$$x(\xi) = X_1(\xi,\zeta) = X_2(\xi,\zeta) = \bar{\xi}X_L,$$

$$Y_1(\xi,\zeta) = Y_1(\xi) = \xi X_L \tan\theta_y$$

$$Y_2(\xi,\zeta) = Y_2(\xi) = \xi X_L(\tan\theta_y + Y_L)$$

$$z(\xi,\zeta) = Z_1(\xi,\zeta) = Z_2(\xi,\zeta) = \xi X_L(\tan\theta_z + Z_L)\bar{\zeta} + \xi X_L \tan\theta_z(1 - \bar{\zeta}),$$

$$\bar{\zeta} = \frac{e^{k_3\zeta} - 1}{e^{k_3} - 1}, \quad \xi = \frac{e^{k_1\bar{\xi}} - 1}{e^{k_1} - 1}. \tag{52}$$

and

$$x = \bar{\xi}X_L,$$

$$y = Y_2(\xi)\bar{r} + Y_1(\xi)(1 - \bar{r}),$$

$$z = z(\xi,\zeta),$$

where

$$\bar{r} = \frac{e^{k_2 r} - 1}{e^{k_2} - 1} \quad .$$

A fundamental constraint of this approach is the representation of the boundaries. The boundaries can be represented as analytical functions or approximate functions based on discrete data from the boundaries. In either case the representation must be in a form where parametric variables which can be normalized to the unit interval are the independent variables. If the parametric independent variables are chosen to be s and t, then for the two boundaries

$$
\begin{aligned}
&X_1(\xi,\eta) \rightarrow X_1(s,t),\\
&Y_1(\xi,\eta) \rightarrow Y_1(s,t),\\
&Z_1(\xi,\eta) \rightarrow Z_1(s,t),\\
&X_2(\xi,\eta) \rightarrow X_2(s,t),\\
&Y_2(\xi,\eta) \rightarrow Y_2(s,t), \qquad (53)\\
&Z_2(\xi,\eta) \rightarrow Z_2(s,t),\\
&0 \le \xi \le 1, \quad s_{min} \le s \le s_{max},\\
&0 \le \eta \le 1, \quad t_{min} \le t \le t_{max}.
\end{aligned}
$$

The choice of parametric variables can vary from problem to problem. A relationship between (ξ,ζ) and (s,t) is

$$
\begin{aligned}
s &= \xi(s_{max} - s_{min}) + s_{min}, \\
t &= \zeta(t_{max} - t_{min}) + t_{min}.
\end{aligned}
\tag{54}
$$

This is a linear relation which maps the unit interval onto the parametric variables. Contraction of the physical grid at the boundaries is accomplished in the same manner as for the internal grid distribution.

$$
\begin{aligned}
&\bar{\xi} = \bar{\xi}(\xi), \quad \frac{d\bar{\xi}}{d\xi} > 0, \\
&\bar{\zeta} = \bar{\zeta}(\zeta), \quad \frac{d\bar{\zeta}}{d\zeta} > 0, \\
&0 \leq \bar{\xi} \leq 1, \quad 0 \leq \xi \leq 1, \\
&0 \leq \bar{\zeta} \leq 1, \quad 0 \leq \zeta \leq 1.
\end{aligned}
\tag{55}
$$

Approximate Boundary-Fitted Coordinate Systems Using Tension Spline Functions

It is often the case that boundaries in a physical domain are described by discrete sets of points. The boundaries may be open or closed. An approximate boundary-fitted coordinate system can be obtained using the technique described and a tension spline function interpolation to the discrete data defining the boundaries. Tension splines are chosen because standard cubic splines and other higher order approximation techniques often result in wiggles in the approximation. Wiggles on a boundary using the technique propagate into the interior grid. The tension parameter embedded in the tension spline approximation to the curve allows control of the "curvedness" of the approximation. A very large magnitude of the tension parameter corresponds to a linear approximation, whereas a very small value corresponds to cubic splines. Tension splines can be applied in two and three dimensions. An example is presented here that is applicable to a two-dimensional mesh.

Using the tension spline technique, a point set on boundary one is defined by $\{x_i,y_i\}_{i=1}^{i=n}$ and on boundary two by $\{x_j,y_j\}_{j=1}^{j=m}$. Approximate arc length is used as a parametric independent variable. The approximate arc length is:

$$s_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} + s_{i-1},$$

$$s_j = [(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2]^{1/2} + s_{j-1},$$

$$i = 1 \ldots n$$

$$j = 1 \ldots m \tag{56}$$

$$s_1 = 0$$

$$0 \leq s_i \leq s_n$$

$$0 \leq s_j \leq s_m.$$

From the computational coordinate system the unit interval $(0 \leq \xi \leq 1)$ must be mapped onto each boundary; that is:

$$\begin{aligned} & s = s(\xi), \\ & 0 \leq \xi \leq 1, \\ & 0 \leq s \leq s_n, \\ & 0 \leq s \leq s_m. \end{aligned} \tag{57}$$

This is accomplished by letting

$$\begin{aligned} & s = \xi s_n \quad \text{on boundary one and} \\ & s = \xi s_m \quad \text{on boundary two.} \end{aligned} \tag{58}$$

The tension spline functions are piecewise continuous hyperbolic functions on each boundary such that

$$\begin{aligned} x = {} & g''(s_\ell)\,\frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\ & + g''(s_{\ell+1})\,\frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\ & + \left[x_\ell - \frac{g''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right) \\ & + \left[x_{\ell+1} - \frac{g''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right) \quad , \end{aligned} \tag{59a}$$

$$
\begin{aligned}
y = {} & h''(s_\ell)\,\frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\
& + h''(s_{\ell+1})\,\frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\
& + \left[y_\ell - \frac{h''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right) \\
& + \left[y_{\ell+1} - \frac{h''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right), \qquad (59b)
\end{aligned}
$$

$\ell = i$ on boundary one,

$\ell = j$ on boundary two,

σ = tension parameter.

The unknowns in these equations are $g''(s_\ell)$ and $h''(s_\ell)$ which are second derivatives at the data points $\{x_\ell, y_\ell\}_{\ell=1}^{\ell=L}$ and are obtained through enforcement of the continuity of the first derivatives at the data points, and the specification of two end conditions. A tridiagonal system of linear equations results for each set of unknowns. The solutions of the tridiagonal systems yield $g''(s_\ell)$ and $h''(s_\ell)$.

A cubic polynomial and the contracting function $\bar{r} = \frac{e^{kr} - 1}{e^{k} - 1}$ provide the relationship between the computational domain and physical domain. The derivatives $\frac{dX_\ell}{d\eta}$ and $\frac{dY_\ell}{d\eta}$ are:

$$
\begin{aligned}
\frac{dX_\ell}{d\eta} &= K\,\frac{dY_\ell}{ds}, \\
\frac{dY_\ell}{d\eta} &= -K\,\frac{dX_\ell}{ds}.
\end{aligned} \qquad (60)
$$

By defining a grid with constants $\Delta\xi$ and $\Delta\eta$ in the computational domain a corresponding grid is explicitly defined in the physical domain.

An example of an airfoil grid is presented (Fig. 14). Data points on each boundary, magnitude of the normal derivative, and contract parameter values define the grids. Boundary data for the airfoil is shown in Fig. 15. Also, for closed boundaries such as the airfoil, periodic conditions are applied in the spline functions.

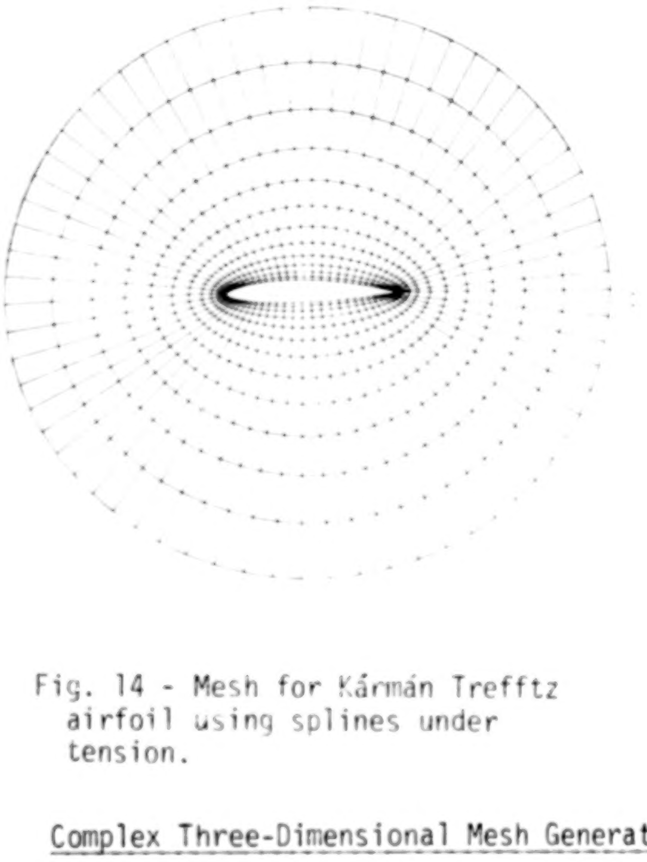


Fig. 14 - Mesh for Kármán Trefftz airfoil using splines under tension.

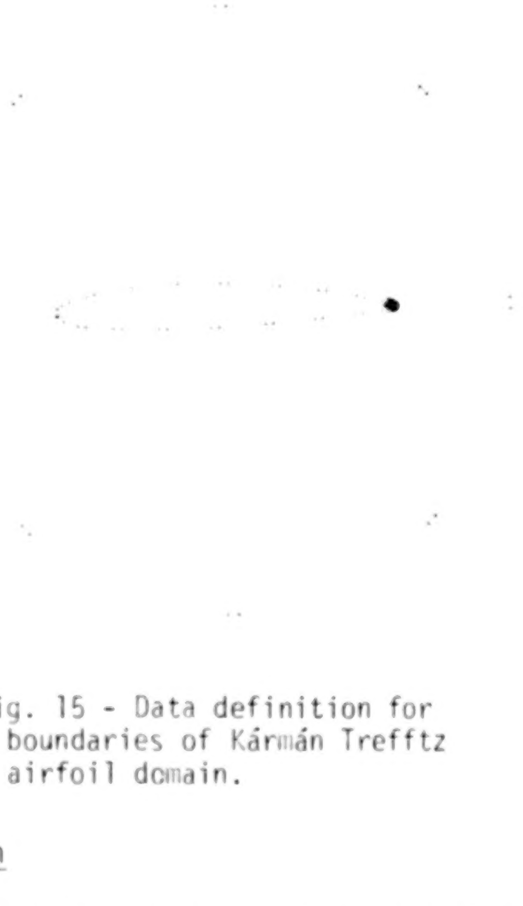


Fig. 15 - Data definition for boundaries of Kármán Trefftz airfoil domain.

Complex Three-Dimensional Mesh Generation

The development of three-dimensional meshes where mesh lines are free to move in all three coordinate directions is extremely difficult. The reader is directed to reference 9 for examples of such unconstrained three-dimensional meshes. An expedient approach for complex three-dimensional geometries is to attempt to simplify the problem by rendering the three-dimensional geometry quasi two-dimensional. This is the essence of the frustrum pyramid mesh previously presented. Also when there is axis-symmetry in a problem, two-dimensional rendering of the geometry is possible. This is demonstrated with the spike-nosed configuration shown in Fig. 16. Figure 17 shows an algebraic generated mesh for one plane of the geometry. The mesh is made three dimensional by rotating the mesh in movements about the axis of symmetry.

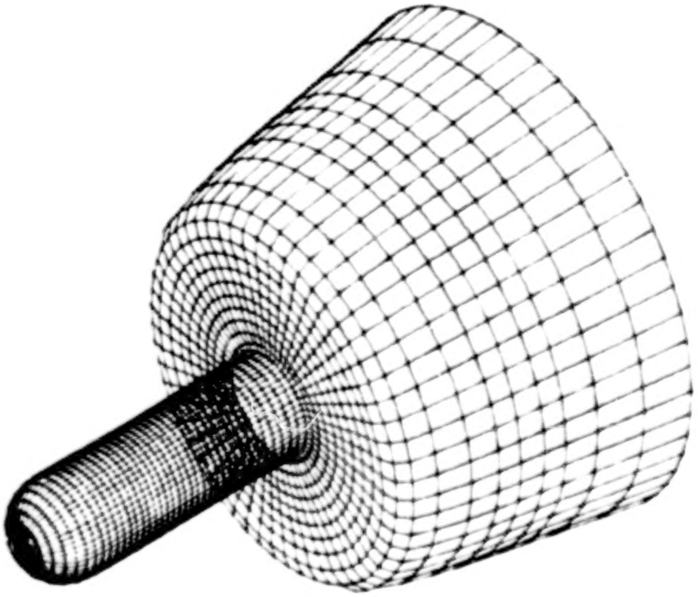


Fig. 16 - Surface for a spike-nosed body.

For aircraft surfaces the problem is more difficult. There are, however, several good software packages ([12] is a good example) for surface definition of aircraft aerospacecraft geometries. In [12]

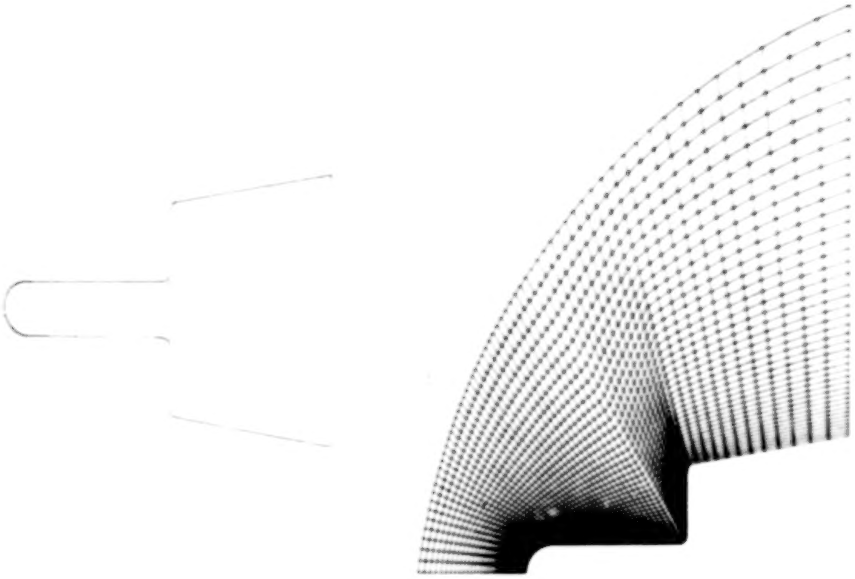


Fig. 17 - Mesh for spike-nosed body.

the Harris geometry is used to establish coordinates for Coon's surface definition of a configuration. Spline functions are used to compute the derivatives for the Coon's surface definition. Figure 18 shows the input description for a wing-fuselage configuration where the wing and fuselage are defined separately. Figure 19 shows an enriched definition of the configuration using the Coon's surfaces. A part of the available software described in [12] is the ability to compute the intersection of an arbitrarily defined plane and the surface definition. For the configuration shown in Fig. 19 planes perpendicular to the x axis and at a constant increment in the x direction are shown in Fig. 20. If an outside boundary is defined the corresponding intersection with the

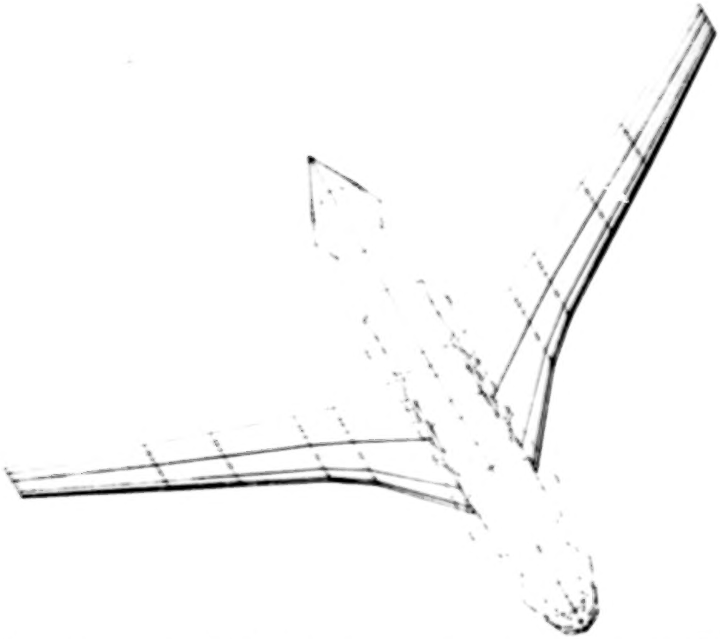


Fig. 18 - Data definition for a wing-fuselage configuration.

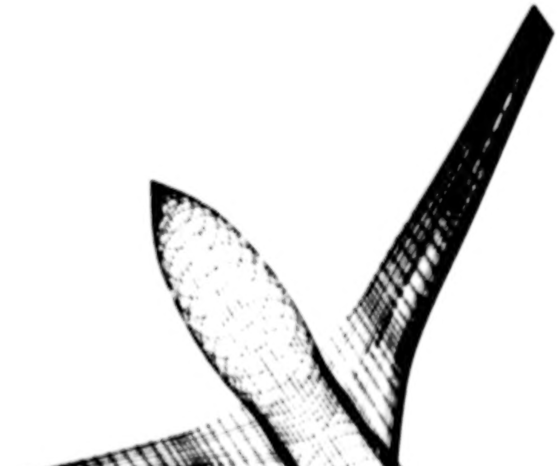


Fig. 19 - Enriched surface definition for a wing-fuselage configuration.

Fig. 20 - Planar intersections with a wing-fuselage configuration.

planes is computable. With an inside and outside boundary the techniques previously described can be employed in two dimensions. Complexities of multiconnected regions is introduced but this can be attacked with branch cuts (Fig. 21).

Fig. 21 - Branch cuts for multi-connected sections.

Another attack on complex three dimensional problems is to define several computational domains (Fig. 22) with mutual boundaries. The mapping

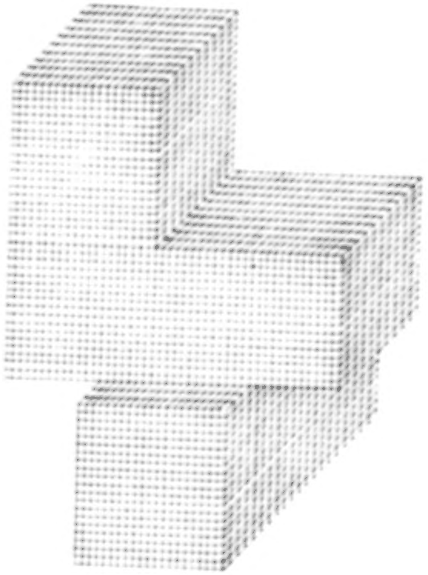


Fig. 22 - Multiple computational domains with mutual boundaries.

function transforms the computational domains to parts of the physical domain with mutual intersections.

Conclusions

Algebraic methods provide precise controls for mesh generation. Methodologies for mesh construction can be based on a parameterized description of surfaces which consist of bounding surfaces and intermediate control surfaces. The surface locations determined by the respective surface parameterizations determine the nature of the transverse coordinate curves which connect the bounding surfaces. Relative to uniform conditions, precise control over the mesh placement in the physical domain can be accomplished by embedding distribution control functions in the surface parameterizations or in the transverse direction. Complex bounding topologies, especially in three dimensions, cause mesh construction difficulties. It is proposed that whenever feasible, the complex topology be simplified such as rendering the geometry quasi two-dimensional. As a final remark, precise controls are one of the major advantages of algebraic methods: they give the capability to prescribe specific desirable and helpful mesh formations.

References

1. Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," *J. Comp. Phys*. 33 (1979), pp. 118-150.

2. Eiseman, P. R., "A Coordinate System for a Viscous Transonic Cascade Analysis," *J. Comp. Phys*. 26 (1978), pp. 307-338.

3. Smith, R. E. and Weigel, B. L., "Analytical and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation," AIAA Paper 80-0192 (1980).

4. Moretti, G. and Abbett, "A Time-Dependent Computational Method for Blunt Body Flows," AIAA J., Vol. 4, No. 12 (1966) pp. 2136-2141.

5. Eiseman, P. R., "Geometric Methods in Computational Fluid Dynamics," von Karman Institute Lecture Series and ICASE Report 80-11 (1980).

6. Eiseman, P. R., "Coordinate Generation with Precise Controls," Seventh International Conference on Numerical Methods in Fluid Dynamics, held at Stanford Univ. and NASA Ames in June 1980.

7. Eiseman, P. R., "Coordinate Generation with Precise Controls Over Mesh Properties." ICASE Report 80-30 (1980).

8. Gordon, W. J. and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," *Intl. J. Num. Meth. in* Engr. 7 (1973), pp. 461-477.

9. Eriksson, L., "Three-Dimensional Spline-Generated Coordinate Transformations for Grids Around Wing-Body Configurations." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 17 of this compilation.)

10. Forsey, C. R., Edwards, M. G., and Carr, M. P., "An Investigation Into Grid Patching Techniques." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 18 of this compilation.)

11. Anderson, P. G., Spradley, L. W., "Finite Difference Grid Generation by Multivariate Blending Function Interpolation." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 6 of this compilation.)

12. Craidon, C. [illegible] "A Computer Program for Fitting Smooth Surfaces to an Aircraft Configuration and Other Three-Dimensional Geometries," NASA TMX-3206, June 1975.

Generation of Boundary and Boundary-Layer Fitting Grids[*]

C. M. Ablow and S. Schechter
SRI International
Menlo Park, CA 94025

ABSTRACT

The details of extended physical processes, such as the gas dynamic flow over an airfoil, the reactive flow through a combustor, or the electric field in a multi-contact transistor, are understood by solving the differential equations of a mathematical model of the process. The accuracy of finite difference methods for the numerical solution of the equations is increased if the underlying mesh fits the region boundaries and is closely spaced in regions where the solution is rapidly varying. Automatic methods for producing a satisfactorily adjusted mesh have been developeed for one-dimensional problems. In one simple, effective scheme of this kind the unknown function and the distribution of mesh modes are found simultaneously, the nodes being placed so that they correspond to points uniformly spaced on the solution curve.

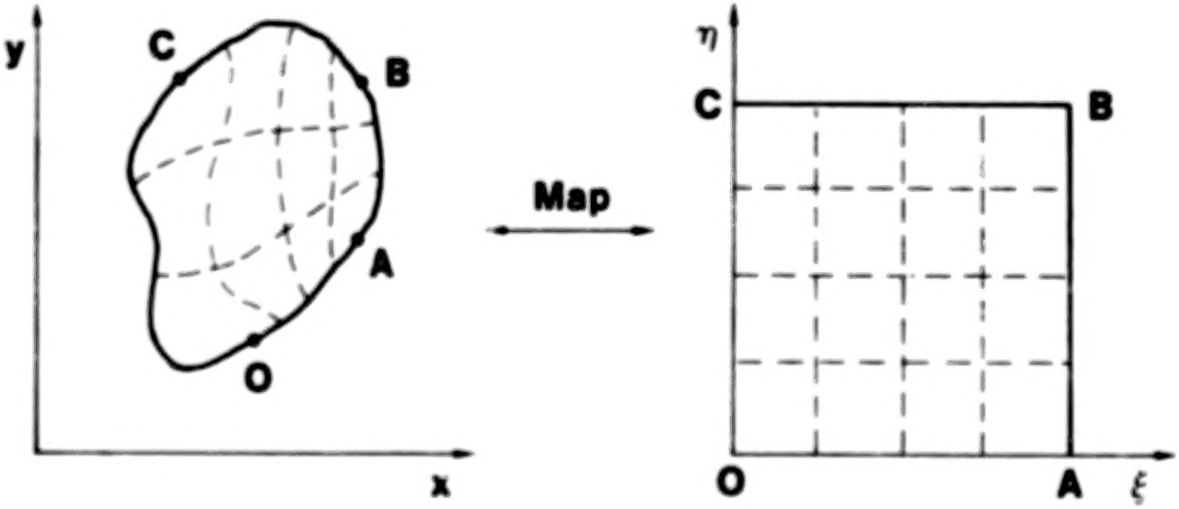
In a two-dimensional generalization, the nodes correspond to points equally spaced on the solution surface in two directions that are as nearly orthogonal as possible. Examples of such meshes are shown for given surfaces in the figures. The meshes fit the circular boundaries and come closer together where the given surface is steeper.

---

[*] Support by the Air Force Office of Scientific Research is gratefully acknowledged.

GENERAL GRID GENERATION METHOD

Map problem domain onto unit cube in computational coordinate space. Uniform, rectangular grid in computer coordinates gives curvilinear grid in original domain.



Choose map to reduce truncation error of finite difference solution scheme for problem.

Problem: Find $z(x,y)$ so that

$$(^*) \quad z_{xx} + z_{yy} = \rho(x,y) \text{ in } x^2 + y^2 < 1.$$
$$z = b(x,y) \text{ on } x^2 + y^2 = 1.$$

with ρ and b given functions.

Change coordinates: $(x,y) \rightarrow (\xi, \eta)$

$$(^*) \quad \frac{\partial}{\partial \xi} \frac{g z_\xi - f z_\eta}{j} + \frac{\partial}{\partial \eta} \frac{e z_\eta - f z_\xi}{j} = j\rho$$

$$ds^2 \equiv dx^2 + dy^2 = e d\xi^2 + 2f d\xi d\eta + g d\eta^2$$

$$e = {x_\xi}^2 + {y_\xi}^2$$

$$f = x_\xi x_\eta + y_\xi y_\eta$$

$$g = {x_\eta}^2 + {y_\eta}^2$$

$$j = (eg - f^2)^{1/2}$$

Truncation error of centered finite-difference approximation

$$= \frac{1}{12j} \Big[g z_{\xi\xi\xi\xi} \Delta\xi^2 + e z_{\eta\eta\eta\eta} \Delta\eta^2 - 4f(z_{\xi\xi\xi\eta} \Delta\xi^2 + z_{\xi\eta\eta\eta} \Delta\eta^2 \Big]$$

Choose boundary-fitting map to

(1) Minimize $(f/j)^2$

(2) Reduce errors in separate ξ and η directions.

One-dimensional monitor function* methods are satisfactory for (2). Boundary adjustment of map used for (1).

---

*A. B. White, Jr., SIAM J Num Anal 16 (1979)
C. M. Ablow and S. Schechter, J. Comp Physics 27 (1978)

One-dimensional monitor function takes equally spaced values at mesh modes.

Simplest, effective, problem-dependent monitor is distance on solution surface:

$\frac{ds}{d\xi}$ is to be constant as ξ varies

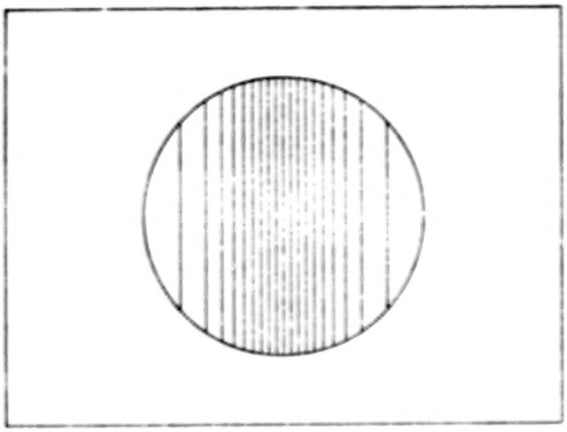
(1) $\frac{\partial}{\partial \xi}(x_{\xi}^{2} + y_{\xi}^{2} + z_{\xi}^{2}) = 0$

Same for η direction

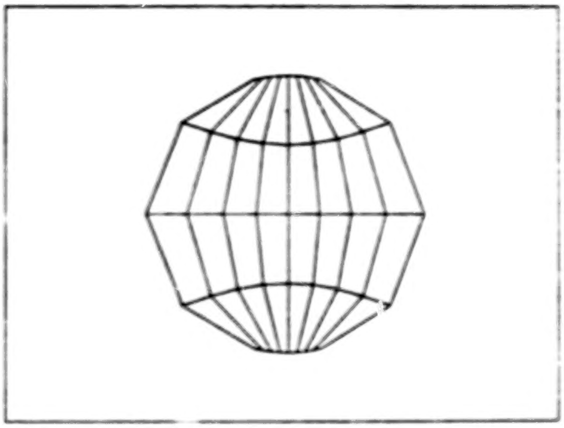
(2) $\frac{\partial}{\partial \eta}(x_{\eta}^{2} + y_{\eta}^{2} + z_{\eta}^{2}) = 0$

(1) and (2) plus given differential equation (*) for z determine solution when corner points O,A,B,C have been chosen. Corners are moved to minimize $\sum (f/j)^2$

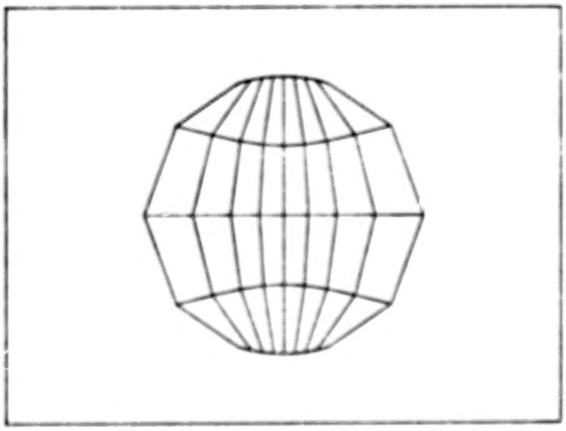
Examples show grids found from (1) and (2) for various given functions z having regions of sharp variation (boundary layers). In practice, function z and the grid mapping would be found by simultaneous solution of the complete set of differential equations (1), (2), and (*).



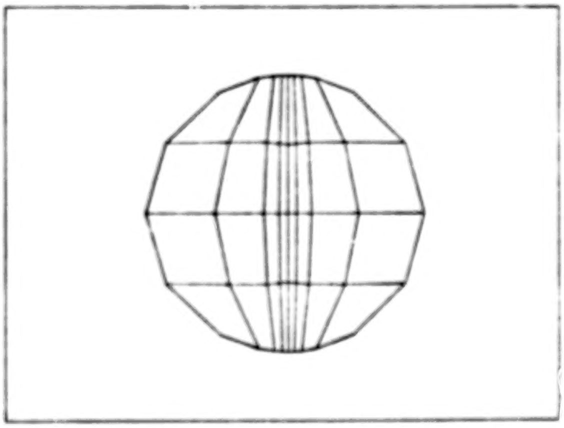
Level lines for
z = tanh px
in unit circle.



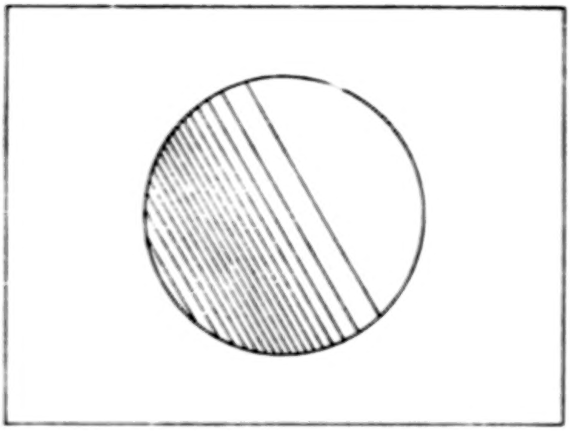
Equidistant mesh
p = 1.0



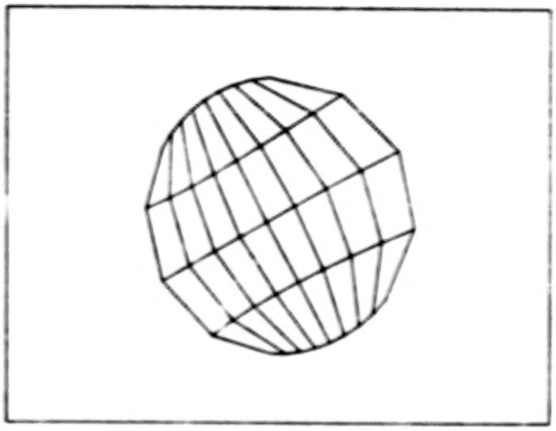
Equidistant mesh
p = 2.0



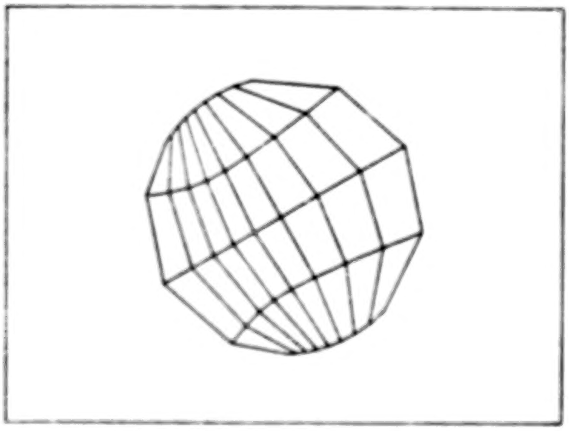
Equidistant mesh
p = 3.0



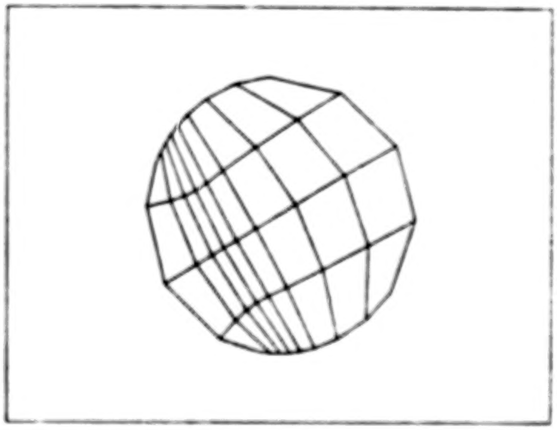
Level lines for $z =$ $\tanh[p(x \cos 30 + y \sin 30) + 0.5]$



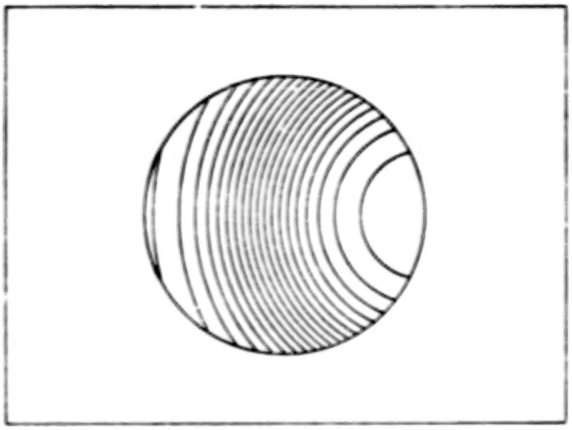
Equidistant mesh
$p = 1.0$



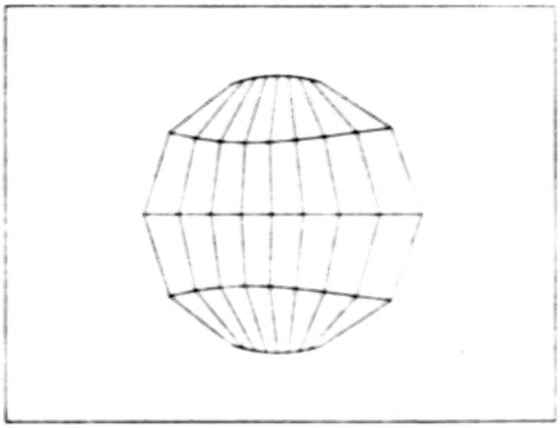
Equidistant mesh
$p = 2.0$



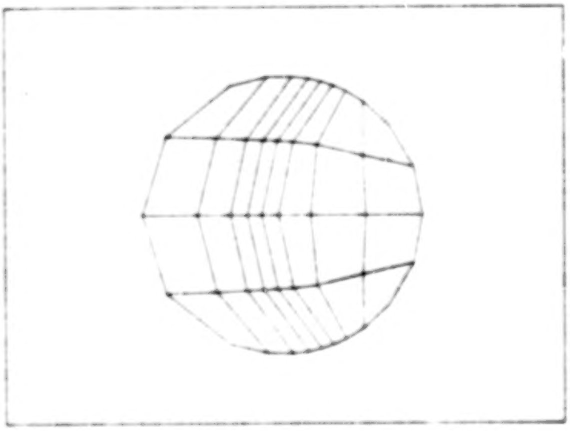
Equidistant mesh
$p = 4.0$



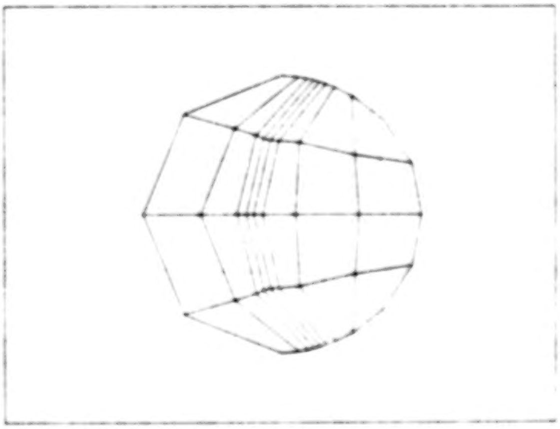
Level lines for
$z = \tanh p\,(R - 1.2)$
$R^2 = (x - 1)^2 + y^2$



Equidistant mesh
$p = 1.0$



Equidistant mesh
$p = 4.0$



Equidistant mesh
$p = 8.0$

BLANK PAGE

BLANK PAGE

AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS

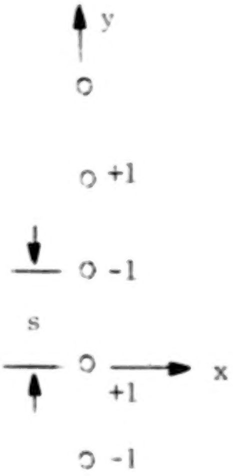
John J. Adamczyk
NASA Lewis Research Center

ABSTRACT

Accurate and efficient numerical simulation of flows through turbomachinery blade rows depends on the topology of the computational grids. These grids must reflect the periodic nature of turbomachinery blade row geometries and conform to the blade shapes. Three types of grids can be generated that meet these minimal requirements: (1) through-flow grids, (2) O-type grids, and (3) C-type grids. This paper presents a procedure which can be used to generate all three types of grids. The resulting grids are orthogonal and can be stretched to capture the essential physics of the flow. In addition, a discussion is also presented detailing the extension of the generation procedure to three-dimensional geometries.

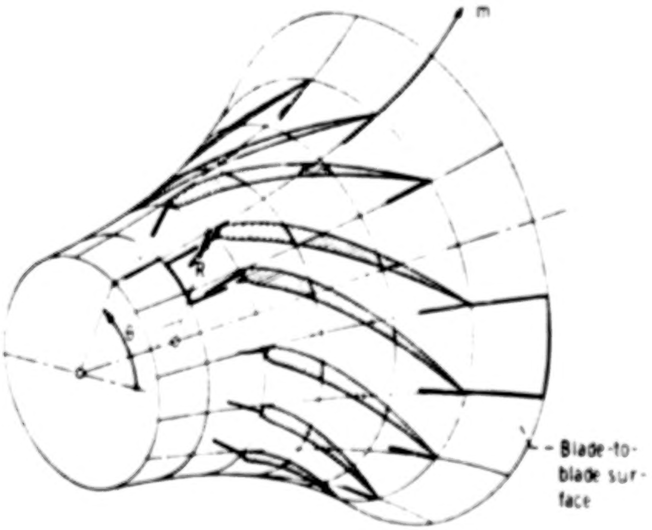
BLADE GEOMETRY AND COORDINATES

The development of the grid generation procedure begins by considering the electrostatic potential field generated by an infinite linear array of point charges in a two-dimensional space. The density of the charges is assumed to alternate between plus and minus 1. The mathematical expression for the complex electrostatic field is given by equation (1), where $K(z - z_0)$ is the complex potential and $i = \sqrt{-1}$. For a Cartesian space $z = x + iy$, z_0 is the location of the zeroth charge and s is the physical distance between charges. For a blade row whose geometry is given on a blade-to-blade surface of revolution, $z = m + i\theta$, where m is the meridional distance and θ is the angular position; s then represents the angular distance between charges.



Array of point charges

$$K(z - z_0) = \sum_{n=-\infty}^{\infty} (-1)^n \ln(z - z_0 - ins) \tag{1}$$



Blade-to-blade surface of revolution, showing m - θ coordinates.

DEVELOPMENT OF EQUATION FOR POTENTIAL FIELD SURROUNDING THE BLADE

The field equation for the array of point charges is expressed in closed form in equation (2). Note that this expression is periodic in either the y- or θ-direction with period 2s. The potential field generated by distributing the fundamental solution (eq. (2)) over the surface of a blade is given by equation (3), where $\gamma(z_o)$ is the source density distribution on the blade surface. It is required that the real part of ω (that is, ξ) be equal to 1 on the blade surface (eq. (4)). At large distances upstream or downstream of the cascade, $\xi(z)$ is assumed to approach zero. This additional requirement is expressed by equation (5). Equation (3) evaluated on the blade surface forms a singular integral equation, equation (6), for the source density. Once γ is known, the potential field surrounding the blade can be computed by direct integration of equation (3).

$$K(z - z_o) = \sum_{n=-\infty}^{\infty} (-1)^n \ln(z - z_o - ins) = \ln \tanh \frac{(z - z_o)\pi}{2s} \qquad (2)$$

$$\omega(z) = \int_L \gamma(z_o)K(z - z_o)dz_o \qquad (3)$$

$$\text{Real } \omega(z) = \xi(z) \equiv 1 \qquad (z \in L) \qquad (4)$$

$$\text{Im} \int_L \gamma(z_o)dz_o = 1 \qquad (5)$$

$$1 = \text{Real}\left[\int_L \gamma(z_o)K(z - z_o)dz_o\right] \qquad (z \in L) \qquad (6)$$

DEVELOPMENT OF EQUATIONS FOR SOURCE DENSITY DISTRIBUTION

The singular integral equations for γ can be solved by paneling methods similar to those used in solving potential flow problems in fluid mechanics. To employ these procedures, one first factors out of the integral equation its singular behavior. (For eq. (6) the factored form is expressed by eq. (7).) Next the blade surface L is divided into a series of segments. Over each of these segments $\gamma(z_0)$ and $\ln\left[\frac{2s}{(z - z_0)\pi} \tanh (z - z_0) \frac{\pi}{2s}\right]$ are approximated by polynomials in z_0. For the cases examined to date it was found that these terms could be approximated by a constant equal to their value at the midpoint of each segment. (This approximation assumes the length of the segment could be made smaller than the scale of the local geometric blade features.) Thus the singular part of equation (7) integrated over a segment is approximated by equation (8), while the regular part is approximated by equation (9). The auxiliary condition (eq. (6)) is approximated over a segment by equation (10). Upon introducing these approximations into equation (7) and restricting the value of z_0 to the midpoint of each segment, a system of linear algebraic equations is obtained from which $\gamma(z_0)$ can be determined.

$$1 = \text{Real}\left[\int_L \gamma(z_0) \ln \frac{(z - z_0)\pi}{2s}\, dz_0 + \int_L \gamma(z_0) \ln \frac{2s}{(z - z_0)\pi} \tanh \frac{(z - z_0)\pi}{2s}\, dz_0\right] \tag{7}$$

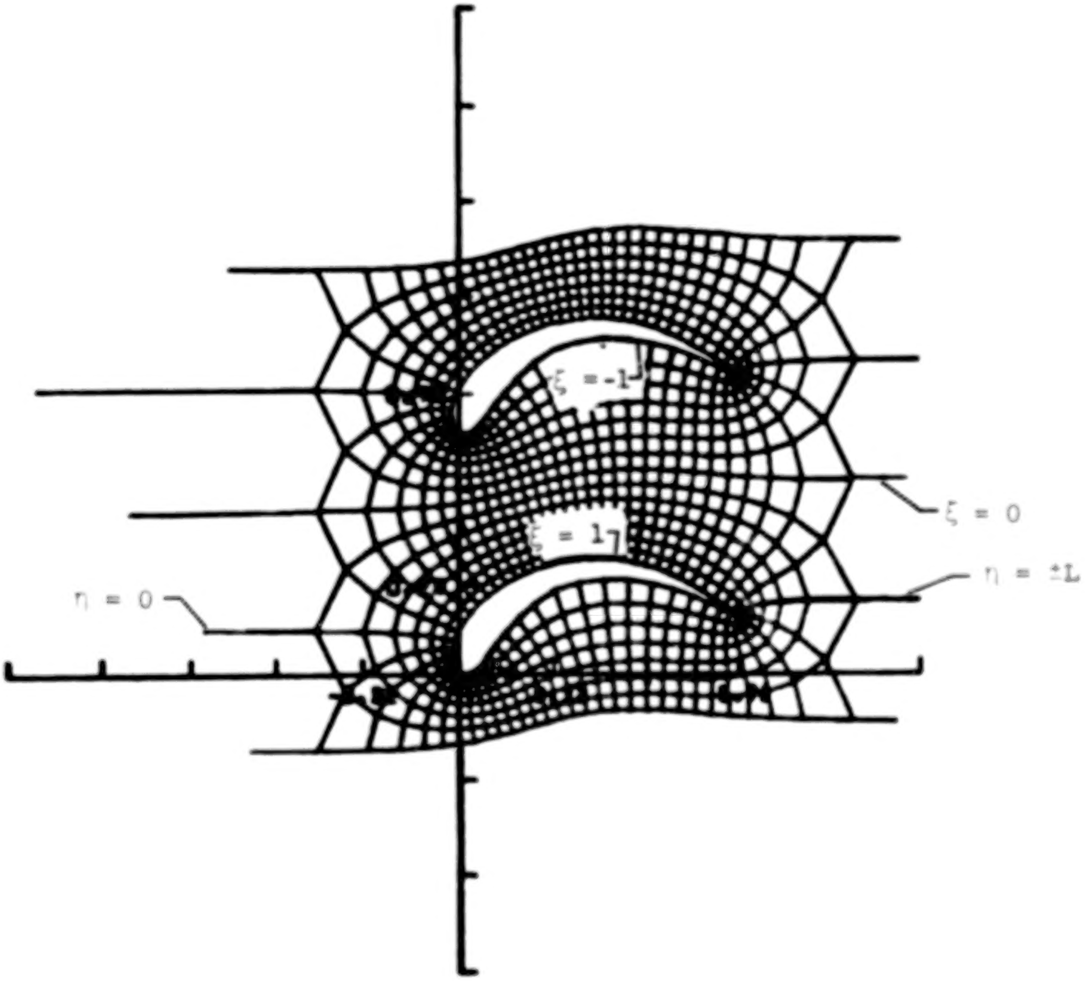
$$\int_a^b \gamma(z_0) \ln \frac{(z - z_0)\pi}{2s}\, dz_0 \approx \gamma(z_0)\Big[(z - b) \ln (z - b) - (z - a) \ln (z - a) + b - a\Big]\Bigg|_{z_0=\frac{b+a}{2}} \tag{8}$$

$$\int_a^b \gamma(z_0) \ln \frac{2s}{(z - z_0)\pi} \tanh \frac{(z - z_0)\pi}{2s}\, dz_0 \approx \gamma(z_0)\left[(b - a) \ln \frac{2s}{(z - z_0)\pi} \tanh \frac{(z - z_0)\pi}{2s}\right]\Bigg|_{z_0=\frac{b+a}{2}} \tag{9}$$

$$\text{Im} \int_a^b \gamma(z_0) dz_0 \approx \text{Im}\Big[\gamma(z_0)(b - a)\Big]\Bigg|_{z_0=\frac{b+a}{2}} \tag{10}$$

BOUNDARIES OF COMPLEX ELECTROSTATIC FIELD SURROUNDING THE BLADE

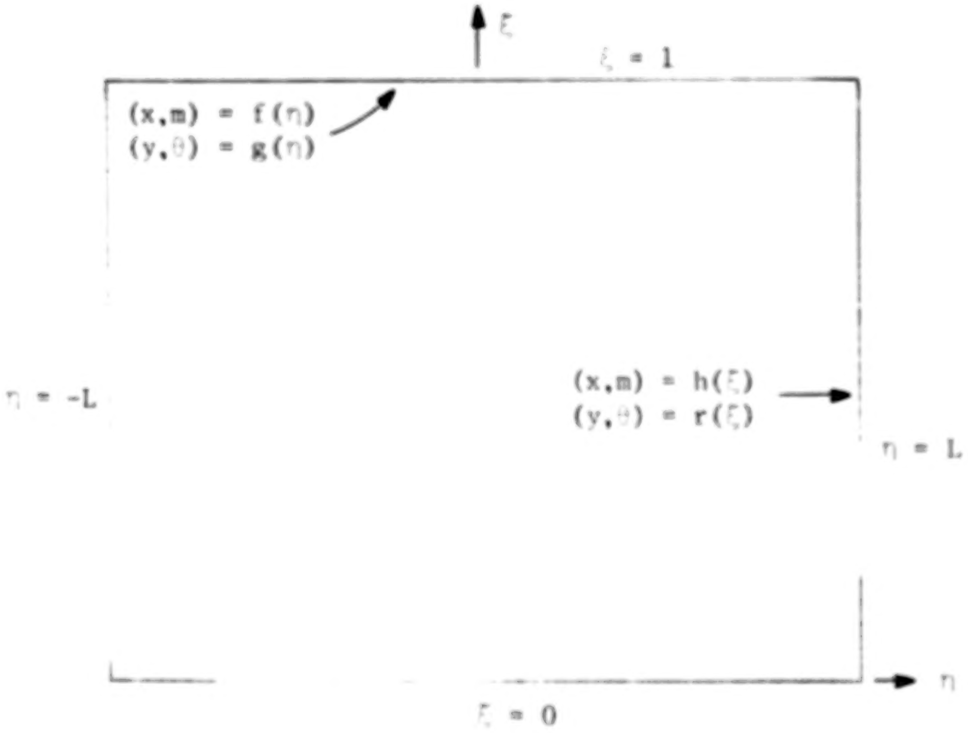
With γ known, the complex electrostatic field surrounding the blade sections can be found. The real part ξ and the imaginary part η of the field form a periodic orthogonal body-fitted coordinate system. The contours ξ = Constant enclose the blade, while the curves η = Constant project from the blade (i.e., $\xi = 1$) to the periodic boundary ($\xi = 0$). The curves extending to upstream and downstream infinity are denoted $\eta = 0$, $\eta = \pm L$, respectively. The locations of these bounding coordinate curves are found using a Newton-Raphson scheme with equation (3) to numerically generate the inverse mapping function. This procedure, however, because of its slow computational speed, was not used to construct the interior grid contours.



INTERIOR GRID

The interior grid points are constructed from the solution of the inverse electrostatic problem in which (x,y) or (m,θ) are specified as functions of ξ or η on the boundaries. The field equation for this is Laplace's equation in terms of (ξ,η) (eq. (11)). The solution of equation (11) which satisfies boundary conditions consistent with the coordinates of the bounding curvilinear curves yields the interior grid geometry. This solution can be obtained by either numerical or analytical procedures.

$$\begin{aligned} \frac{\partial^2(x,m)}{\partial\xi^2} + \frac{\partial^2(x,m)}{\partial\eta^2} &= 0 \\ \frac{\partial^2(y,m)}{\partial\xi^2} + \frac{\partial^2(y,\theta)}{\partial\eta^2} &= 0 \end{aligned} \tag{11}$$



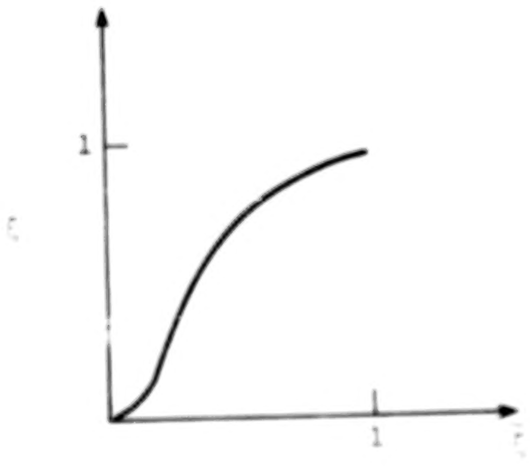
GRID CLUSTERING

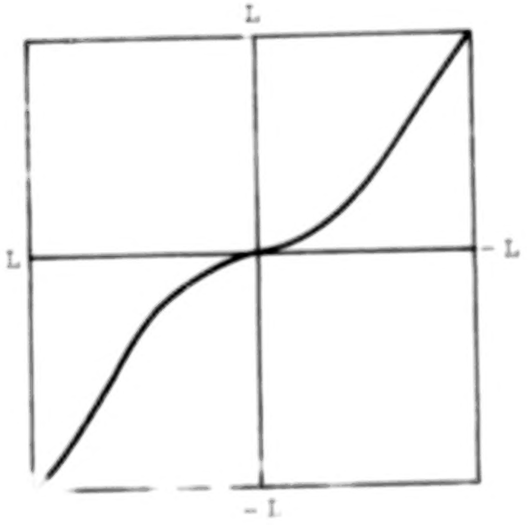
Grid clustering to capture the physics of the flow field can be introduced prior to or after the solution of the inverse problem is obtained. To insure orthogonality, the stretching function used for clustering must be one-dimensional (i.e., eqs. (12)). The ξ transformation can be quite arbitrary. For potential flow computations, a linear transformation is generally used. For viscous flows, one attempts to cluster grid points near the blade surface. An example of a transformation which can be used for this purpose is given by equation (13). The parameters m_o, m_1 control the degree of stretching in the transformation. The clustering of grid points in the η-direction requires special consideration to insure that grid point periodicity is maintained. A grid point located on the periodic boundary at $\eta = \eta_o$ has an image at $\eta = -\eta_o$. In order to maintain this property, the transformation in η must be an odd function of $\bar{\eta}$ over the interval $-L$ to L. A simple transformation which exhibits this behavior is a polynomial in odd powers of $\bar{\eta}$, an example being given by equation (14). The parameters m_o and m_1 in this transformation are again used to control the degree of clustering.

$$\left.\begin{aligned} \xi &= \xi(\bar{\xi}) \qquad && (0 \le \bar{\xi} \le 1) \\ \eta &= \eta(\bar{\eta}) \qquad && (-L \le \bar{\eta} \le L) \end{aligned}\right\} \tag{12}$$

$$\xi = m_o\bar{\xi} + (3 - m_1 - 2m_o)\bar{\xi}^2 + (m_1 + m_o - 2)\bar{\xi}^3 \tag{13}$$

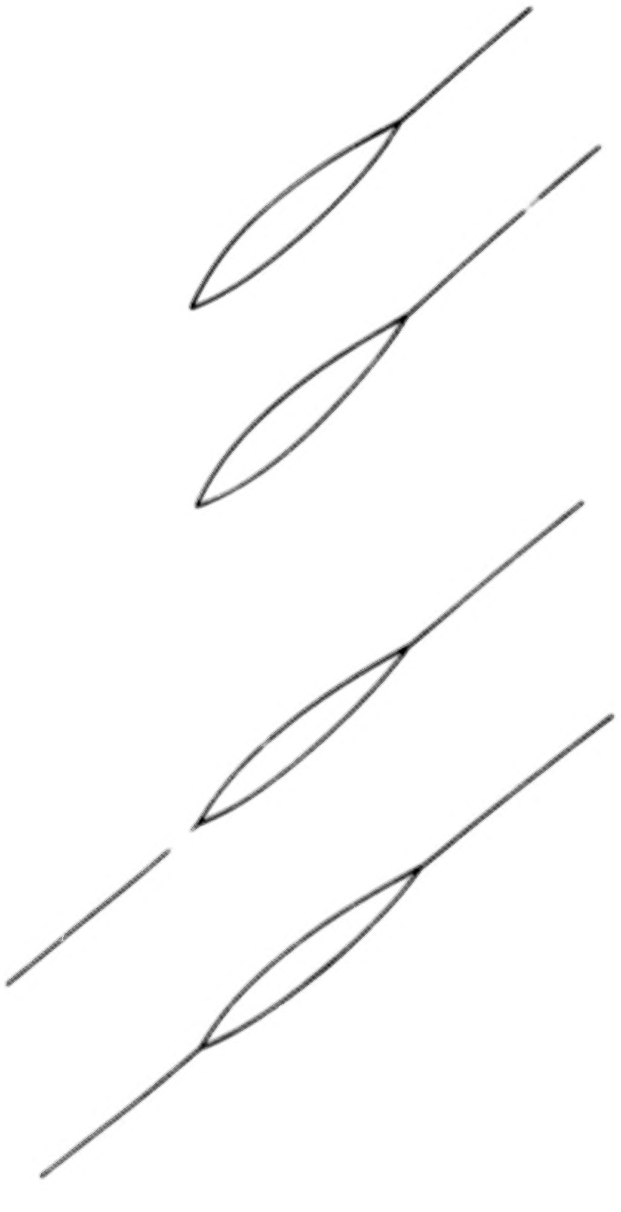
$$\eta = m_o\bar{\eta} + \frac{1}{2L^2}(5 + 4m_o - m_1)\bar{\eta}^3 - \frac{1}{2L^4}(3 - 2m_o - m_1)\bar{\eta}^3 \tag{14}$$





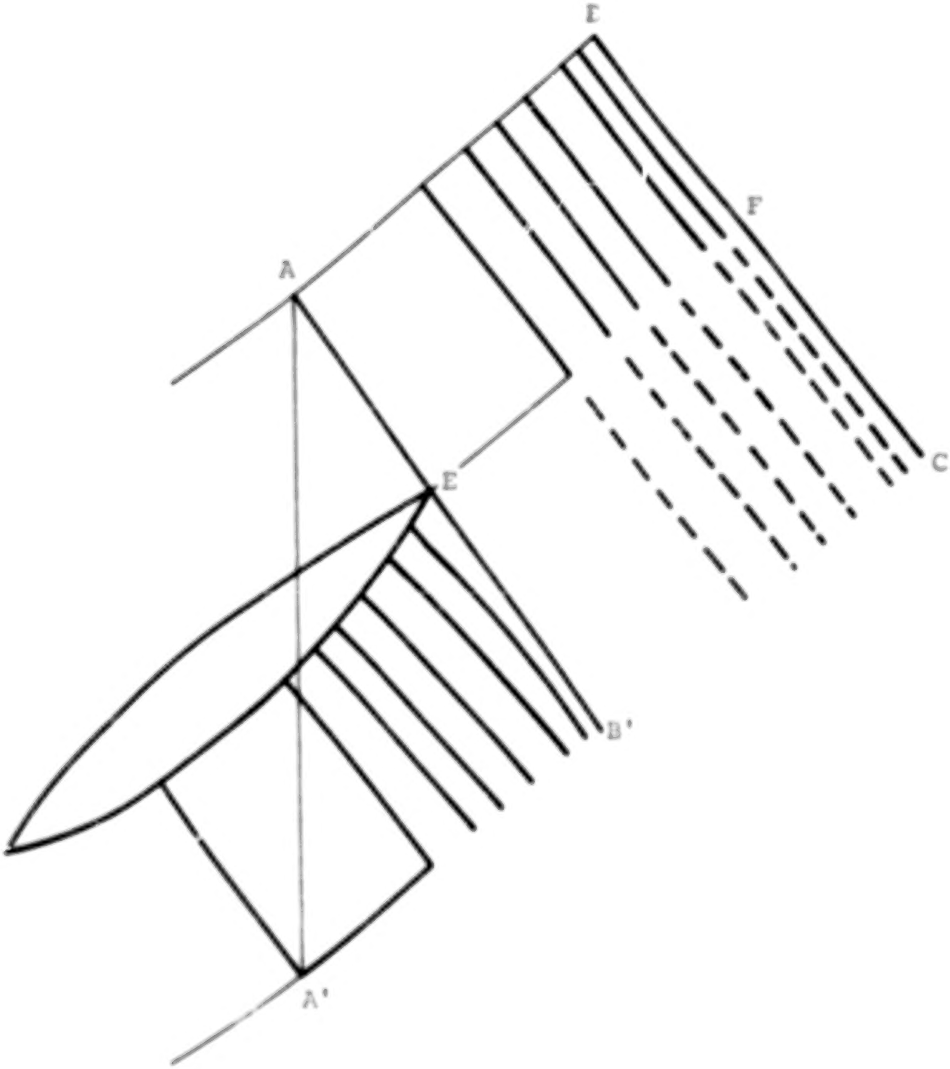
GENERATION OF C-TYPE AND THROUGH-FLOW GRIDS

Thus far the application of the grid generation procedure has been restricted to orthogonal O-type grids. To generate through-flow and C-type grids by the current procedure, the blade contours must be modified by appending slits of zero thickness to their surfaces. For C-type grids, one slit is used. Its origin is generally taken to be the trailing edge of the blade. For through-flow grids, two slits are appended, their origins being the leading and trailing edges of the blade. The shape of these appendages can be quite general. The generation of the grids associated with these modified blade profiles proceeds as in the O-grid procedure outlined above. The generated grids are orthogonal and periodic. In the case of blunt blades, however, they exhibit a singular behavior at the slit attachment point.



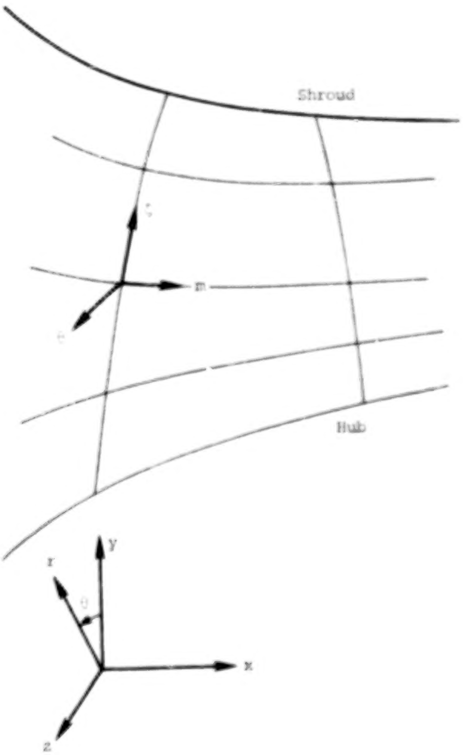
GRIDS FOR CASCADES

For cascades of nonzero stagger, the C-type or through-flow grids generated by the current procedure are discontinuous across the slit. This undesirable property can be corrected by a simple construction. First the two grid lines connecting the upper periodic boundary and the trailing edge and the lower periodic boundary and the trailing edge are found. Next the location and value of η corresponding to the periodic image of A on the lower boundary (i.e., A') and B' on the upper boundary (i.e., B) are determined. The spacing of grid points along the slit between EF is specified, which in turn determines the distribution of η along both sides of the slit. This in turn determines the distribution of η along the periodic boundaries AB and B'C'. The grid points along A'B' are required to be periodic images of the points along AB. This determines the distribution of η along A'B' and the grid geometry up to EB'. The construction of the grid downstream of B'C' proceeds in the same manner as outlined above, with the grid spacing along B'C' defining the clustering pattern. The resulting grids will remain orthogonal and periodic under this construction procedure.



DEVELOPMENT OF THREE-DIMENSIONAL GRIDS

Three-dimensional grids can also be developed by the current procedure. The geometry of a typical blade assembly as viewed in the meridional plane is shown in the accompanying figure. Let the surfaces of revolution describing the hub and shroud be denoted by $r = f_H(x)$, $r = f_S(x)$. A surface of revolution bounded by these limits is given by equation (15). Similarly, let m denote the percentage of distance measured from the leading edge of the blade along the surfaces of revolution (eq. (16)). On a surface of revolution the blade section geometry is given in terms of m and θ, where θ is the angular location around the wheel. On the blade surface of revolution m and θ can be expressed in terms of the coordinates ξ,η of the current orthogonal system. The resulting coordinate system (ζ,ξ,η) will be orthogonal on a ζ-plane and conform to the blade hub and shroud surface.

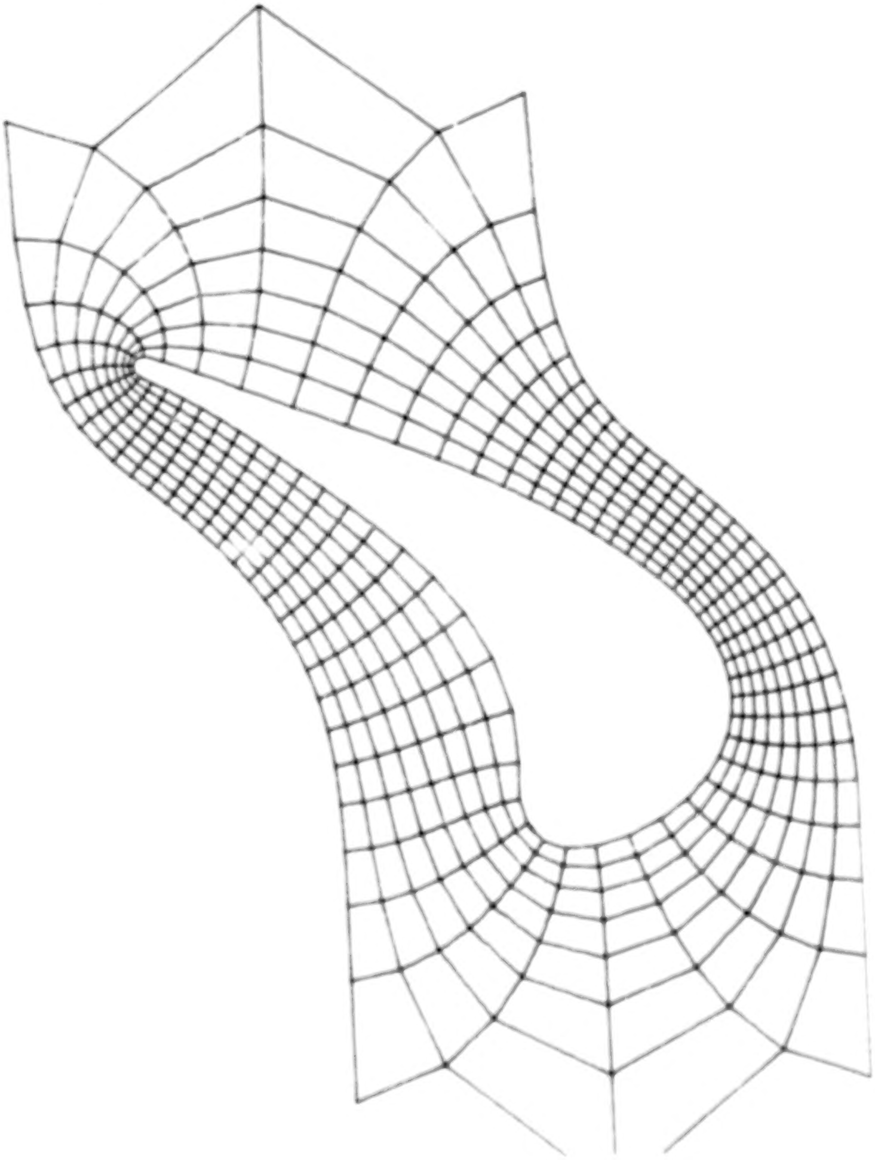


$$\zeta = \frac{r - f_H(x)}{f_S(x) - f_H(x)} \qquad (f_H \leq r \leq f_S) \qquad (15)$$

$$m = \frac{\int_0^{m_\zeta} ds}{\int_0^{L_\zeta} ds} \qquad (16)$$

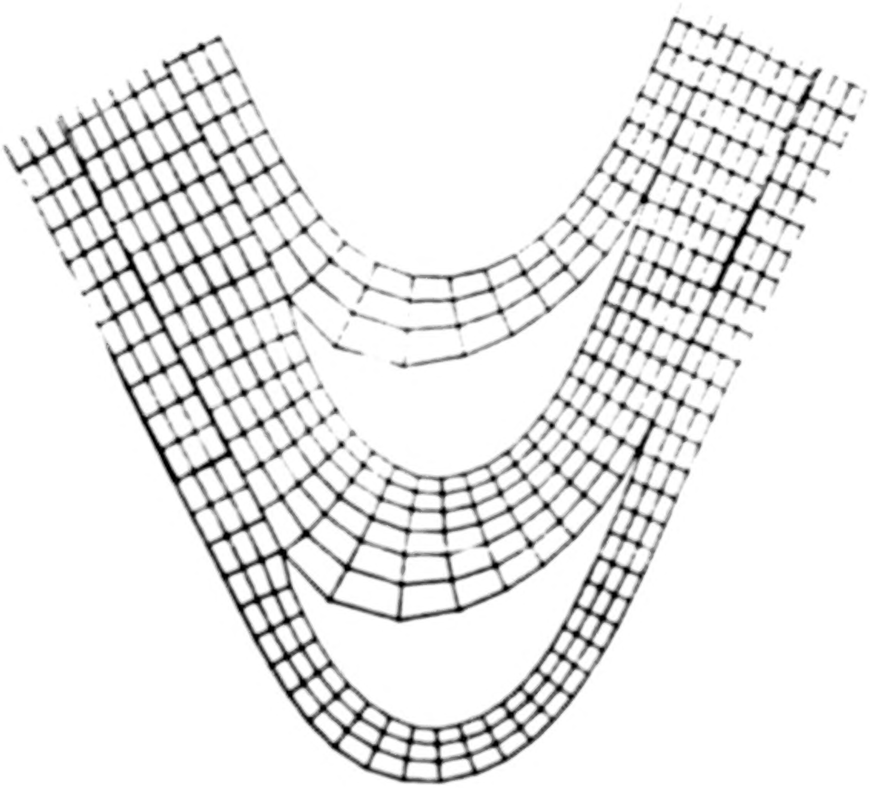
EXAMPLE O-TYPE GRID FOR TURBINE STATOR BLADE

An example of an O-type grid generated by the current procedure is shown on the accompanying figure. The blade is a turbine stator with approximately 90° of turning. No stretching was introduced in developing the grid. In regions of high surface curvature there is a high concentration of grid points, thus permitting accurate resolution of the local flow physics. Far removed from the blade, the concentration of grid points becomes sparse. This results in an economical distribution of grid points in regions of uniform flow. The grid as shown was generated to solve a potential flow problem. For viscous flow, a stretching of the ξ contours would have to be introduced to capture the boundary-layer region. In addition, the η lines would have to be clustered in the trailing-edge region to resolve the viscous wake.



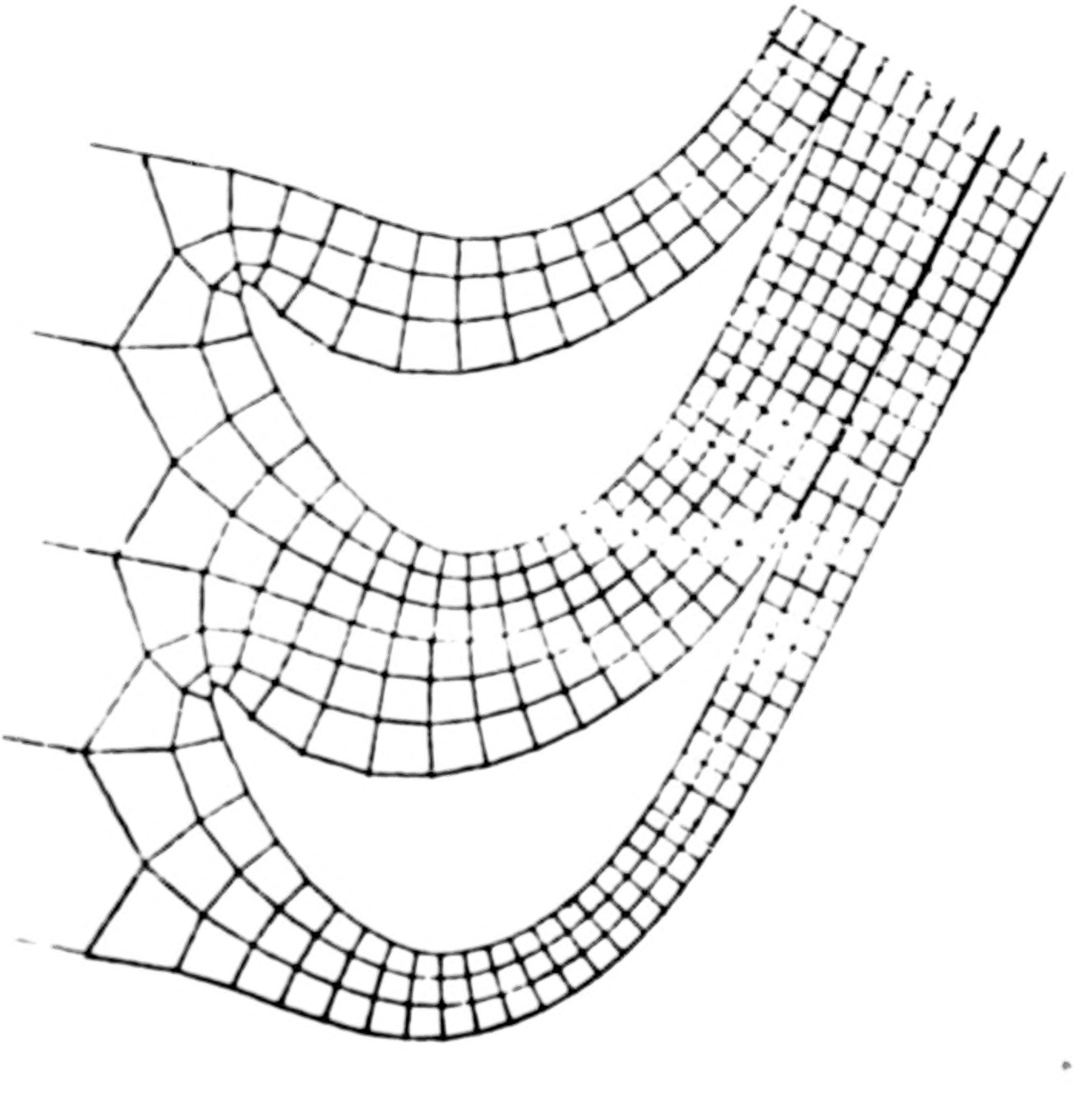
EXAMPLE THROUGH-FLOW GRID FOR A TURBINE ROTOR BLADE

The next example is a through-flow grid for a high-reaction turbine rotor. This grid, as described earlier in the paper, was generated by appending two slits to the blade surfaces. The grid geometry is rectangular and periodic upstream and downstream of the cascade. Across the wake the grid geometry is seen to be discontinuous. The use of such a grid would require special care in transferring flow variables across the slit.



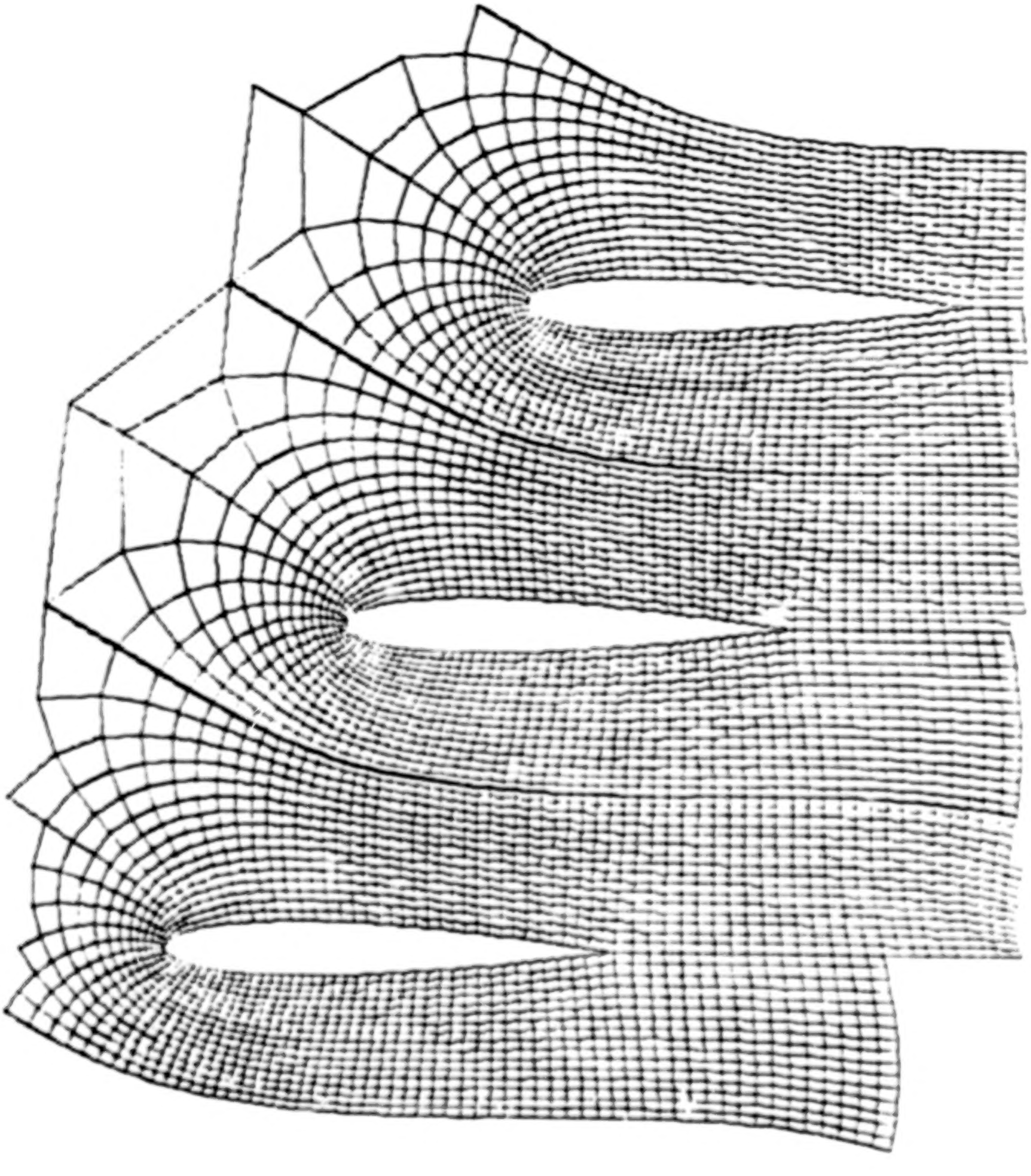
EXAMPLE C-TYPE GRID FOR A TURBINE ROTOR BLADE

The accompanying figure shows a C-type grid for the preceding turbine blade. Upstream of the blade the grid geometry is similar to an O-type grid, while downstream it resembles a through-flow grid. For this grid the construction procedure outlined earlier was used to insure continuous grid geometry across the slit. For potential flow analysis the grid as shown is quite suitable. For viscous analysis a clustering of the grid lines near the blade surface and trailing-edge region would have to be introduced to accurately resolve the flow physics.



EXAMPLE C-TYPE GRID FOR CASCADE OF NACA 0012 AIRFOILS

The final example is a C-type grid for a cascade of NACA 0012 airfoils. This grid was generated by When-Huei Jou of Flow Research. The grid generation procedure as modified by him produces grids which are nearly orthogonal., The slight nonorthogonality is due to stretching functions used to maintain continuous grid geometry across the slit and cascade periodicity. This grid was generated to solve a two-dimensional potential flow problem. It can also be used to solve a three-dimensional potential flow problem, provided the inviscid wake is convected downstream along the slit. For viscous flow calculation a clustering of grid points near the body and wake region would have to be introduced.



NACA 0012 airfoil. Stagger angle 30.00; pitch 0.794; cascade grid 128 × 16.

FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION*

Peter G. Anderson and Lawrence W. Spradley
Lockheed-Huntsville Research & Engineering Center
Huntsville, Alabama

ABSTRACT

The General Interpolants Method (GIM) code solves the multidimensional Navier-Stokes equations for arbitrary geometric domains. The geometry module in the GIM code generates two- and three-dimensional grids over specified flow regimes, establishes boundary condition information and computes finite difference analogs for use in the GIM code numerical solution module. The technique can be classified as an algebraic equation approach.

The geometry package uses multivariate blending function interpolation of vector-values functions which define the shapes of the edges and surfaces bounding the flow domain. By employing blending functions which conform to the cardinality conditions the flow domain may be mapped onto a unit square (2-D) or unit cube (3-D), thus producing an intrinsic coordinate system for the region of interest. The intrinsic coordinate system facilitates grid spacing control to allow for optimum distribution of nodes in the flow domain.

The GIM formulation is not a finite element method in the classical sense. Rather, finite difference methods are used exclusively but with the difference equations written in general curvilinear coordinates. Transformations are used to locally transform the physical planes into regions of unit cubes. The mesh is generated on this unit cube and local metric-like coefficients generated. Each region of the flow domain is likewise transformed and then blended via the finite element formulation to form the full flow domain. In order to treat "completely-arbitrary" geometric domains, different transformation functions can be employed in different regions. We then transform the blended domain to physical space and solve the Cartesian set of equations for the full region. The geometry part of the problem is thus treated much like a finite element technique while integration of the equations is done with finite difference analogs.

*This work was supported, in part, by NASA Langley Contracts NAS1-15341, 15783, and 15795.

BUILDING BLOCK CONCEPT

The development is done in local curvilinear intrinsic coordinates based on the following concepts:

- Analytical regions such as rectangles, spheres, cylinders, hexahedrals, etc., have intrinsic or natural coordinates.
- Complex regions can be subdivided into a number of smaller regions which can be described by analytic functions. The degenerate case is to subdivide small enough to use very small straight-line segments.
- Intrinsic curvilinear coordinate systems result in constant coordinate lines throughout a simply connected, bounded domain in Euclidean space.
- The intersection of the lines of constant coordinates produce nodal points evenly spaced in the domain.
- Intrinsic curvilinear coordinate systems can be produced by a univalent mapping of a unit cube onto the simply connected bounded domain.

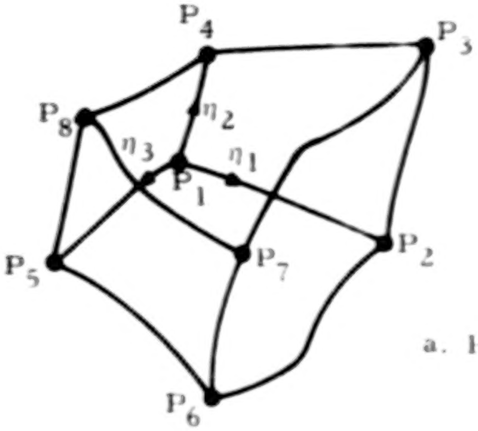
Thus, if a transformation can be found which will map a unit cube univalently onto a general analytical domain, then any complex region can be piecewise transformed and blended using general interpolants.

Consider the general hexahedral configuration shown. The local intrinsic coordinates are η_1, η_2, η_3 with origin at point P_1. The shape of the geometry is defined by

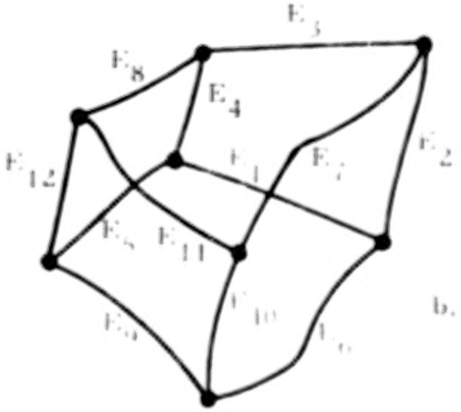
- Eight corner points, $\overline{P}_i$
- Twelve edge functions, $\overline{E}_i$
- Six surface functions, $\overline{S}_i$

This shape is then fully described if the edges and surfaces can be specified as continuous analytic vector functions $\overline{S}_i(x, y, z)$, $\overline{E}_i(x, y, z)$.

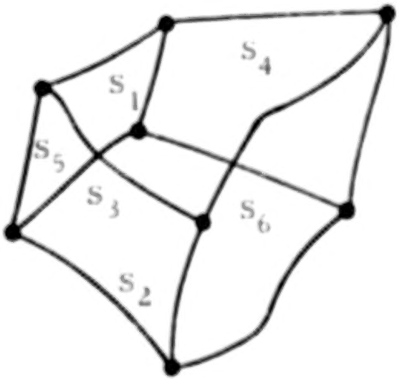
BUILDING BLOCK CONCEPT


P_4
P_3
P_8
η_2
η_3
η_1
P_1
P_5
P_7
P_2
P_6

a. Point Designations


E_3
E_8
E_4
E_{12}
E_1
E_7
E_2
E_5
E_{11}
E_{10}
E_9
E_6

b. Edge Designations


S_4
S_1
S_5
S_3
S_6
S_2

c. Surface Designations

GENERAL INTERPOLANT FUNCTION

Based on the work of Gordon and Hall we have developed a general relationship between physical Cartesian space and local curvilinear intrinsic coordinates. This relation is given by the general trilinear interpolant shown on the adjacent figure.

In this equation, $\underline{\bar{X}}$ vector is the Cartesian coordinates

$$\underline{\bar{X}}(\eta_1, \eta_2, \eta_3) = \begin{bmatrix} x(\eta_1, \eta_2, \eta_3) \\ y(\eta_1, \eta_2, \eta_3) \\ z(\eta_1, \eta_2, \eta_3) \end{bmatrix}$$

and S_i, E_i are the vector functions defining the surfaces and edges, respectively, and $\overline{P}_i$ are the (x, y, z) coordinates of the corner points. Edge and surface functions that are currently included in the GIM code are the following:

- EDGES (Combinations of up to Five Types)

  Linear Segment
  Circular Arc
  Conic (Elliptical, Parabolic, Hyperbolic)
  Helical Arc
  Sinusoidal Segment

- SURFACES (Bounded by Above Edges)

  Flat Plate
  Cylindrical Surface
  Edge of Revolution

This library of available functions is simply called upon piecewise via input to the computer code.

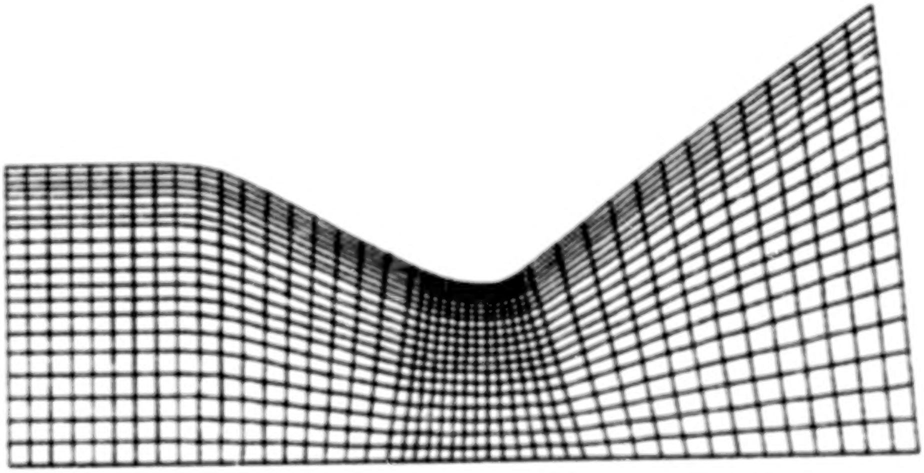
With this transformation, any point in local coordinates η_1, η_2, η_3 can be related to global Cartesian coordinates x, y, z. Likewise any derivatives of functions in local coordinates can be related to that derivative in physical space.

GENERAL INTERPOLANT FUNCTION

$$\begin{aligned}
&\underline{\overline{X}}\,(\eta_1, \eta_2, \eta_3) = \\
&(1-\eta_1)\,\vec{S}_5 + \eta_1\,\vec{S}_6 + (1-\eta_2)\,\vec{S}_2 + \eta_2\,\vec{S}_4 \\
&+ (1-\eta_3)\,\vec{S}_1 + \eta_3\,\vec{S}_3 \\
&- (1-\eta_1)(1-\eta_2)\,\vec{E}_5 - (1-\eta_1)\,\eta_2\,\vec{E}_8 - \eta_1\,(1-\eta_2)\,\vec{E}_6 \\
&- \eta_1\,\eta_2\,\vec{E}_7 - (1-\eta_1)(1-\eta_3)\,\vec{E}_4 - (1-\eta_1)\,\eta_3\,\vec{E}_{12} \\
&- \eta_1\,(1-\eta_3)\,\vec{E}_2 - \eta_1\,\eta_3\,\vec{E}_{10} - (1-\eta_2)(1-\eta_3)\,\vec{E}_1 \\
&- (1-\eta_2)\,\eta_3\,\vec{E}_9 - \eta_2\,(1-\eta_3)\,\vec{E}_3 - \eta_2\,\eta_3\,\vec{E}_{11} \\
&+ (1-\eta_1)(1-\eta_2)(1-\eta_3)\,\vec{P}_1 + (1-\eta_1)(1-\eta_2)\,\eta_3\,\vec{P}_5 \\
&+ (1-\eta_1)\,\eta_2\,(1-\eta_3)\,\vec{P}_4 + (1-\eta_1)\,\eta_2\,\eta_3\,\vec{P}_8 \\
&+ \eta_1\,(1-\eta_2)(1-\eta_3)\,\vec{P}_2 + \eta_1\,(1-\eta_2)\,\eta_3\,\vec{P}_6 \\
&+ \eta_1\,\eta_2\,(1-\eta_3)\,\vec{P}_3 + \eta_1\,\eta_2\,\eta_3\,\vec{P}_7
\end{aligned}$$

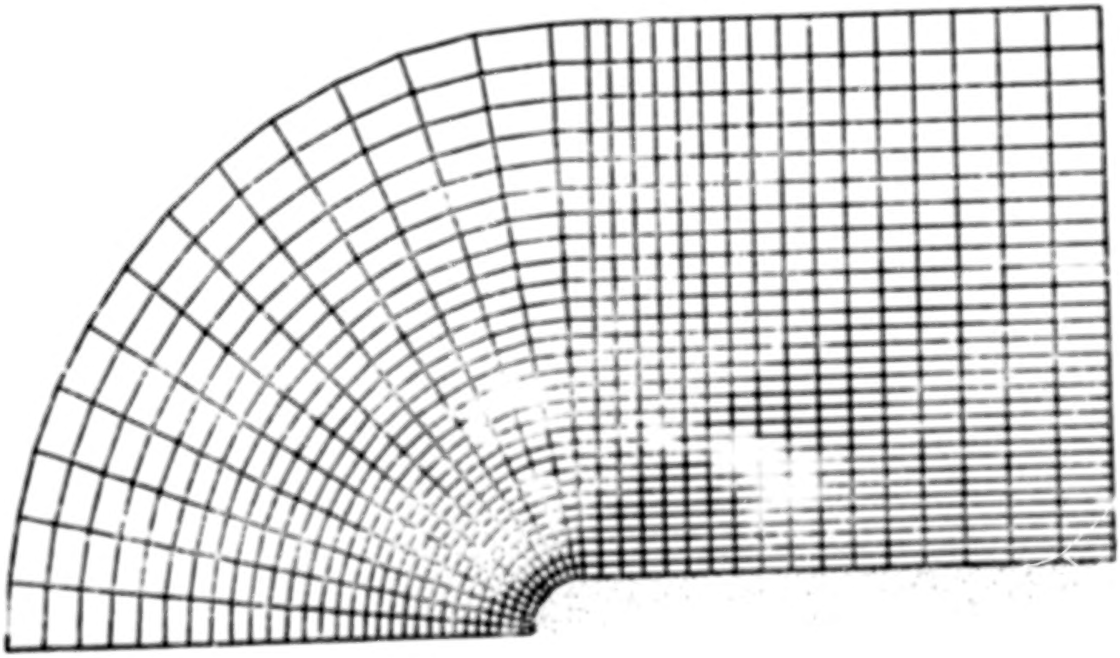
INTERNAL FLOW GRID
(Axisymmetric Rocket Nozzle)

The grid shown was used to compute the flow in a model of the Space Shuttle engine using the GIM code. The mesh is stretched in the radial direction to cluster points near the wall and stretched axially to place points near the throat of the nozzle. Only a portion of the complete grid is shown for clarity and illustration. The grid shows the general format used by the GIM code for internal, two-dimensional flows in non-rectangular shapes.



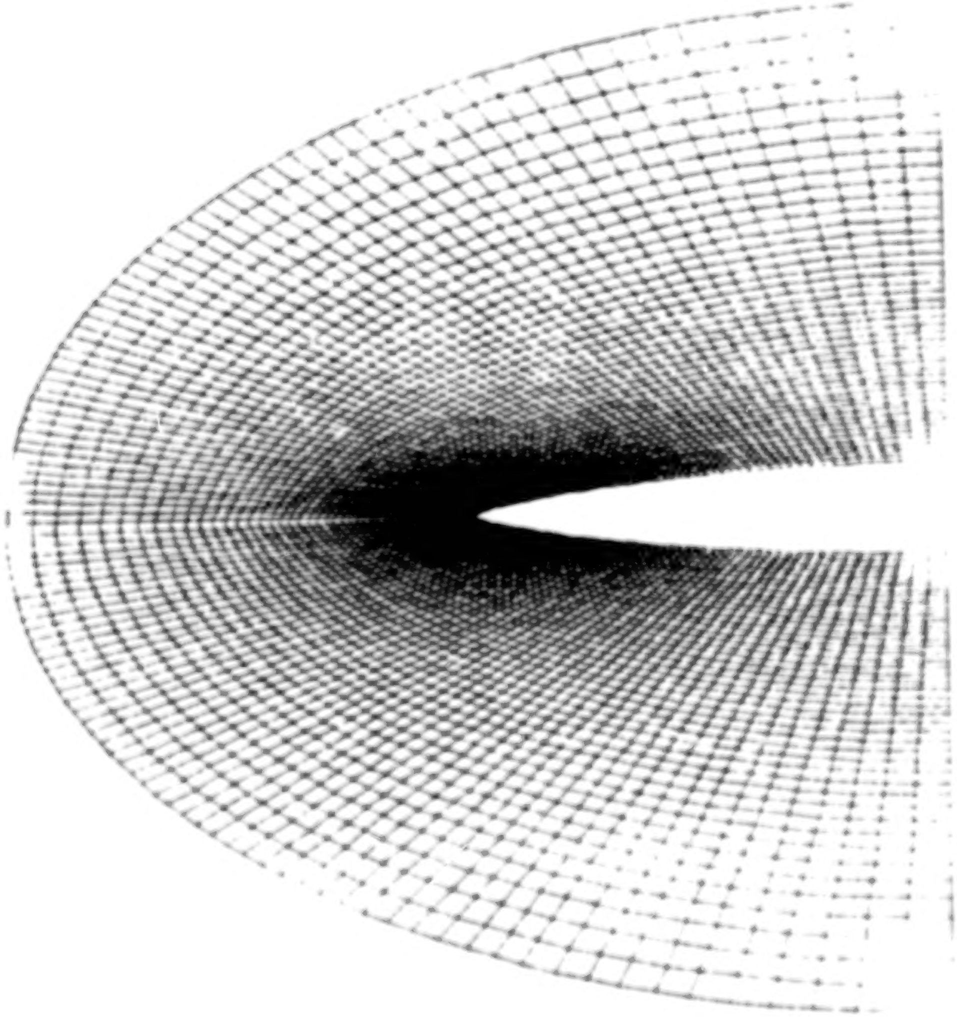
EXTERNAL FLOW GRID
(Two-Dimensional Blunt Body Flow)

This figure shows a polar-like grid used for computing external flow over a blunt body. The body surface is treated inviscidly, and thus does not require an extremely tight mesh. The outer boundary is the freestream flow. The grid illustrates the GIM code technique for two-dimensional external flows using a polar-like coordinate system.



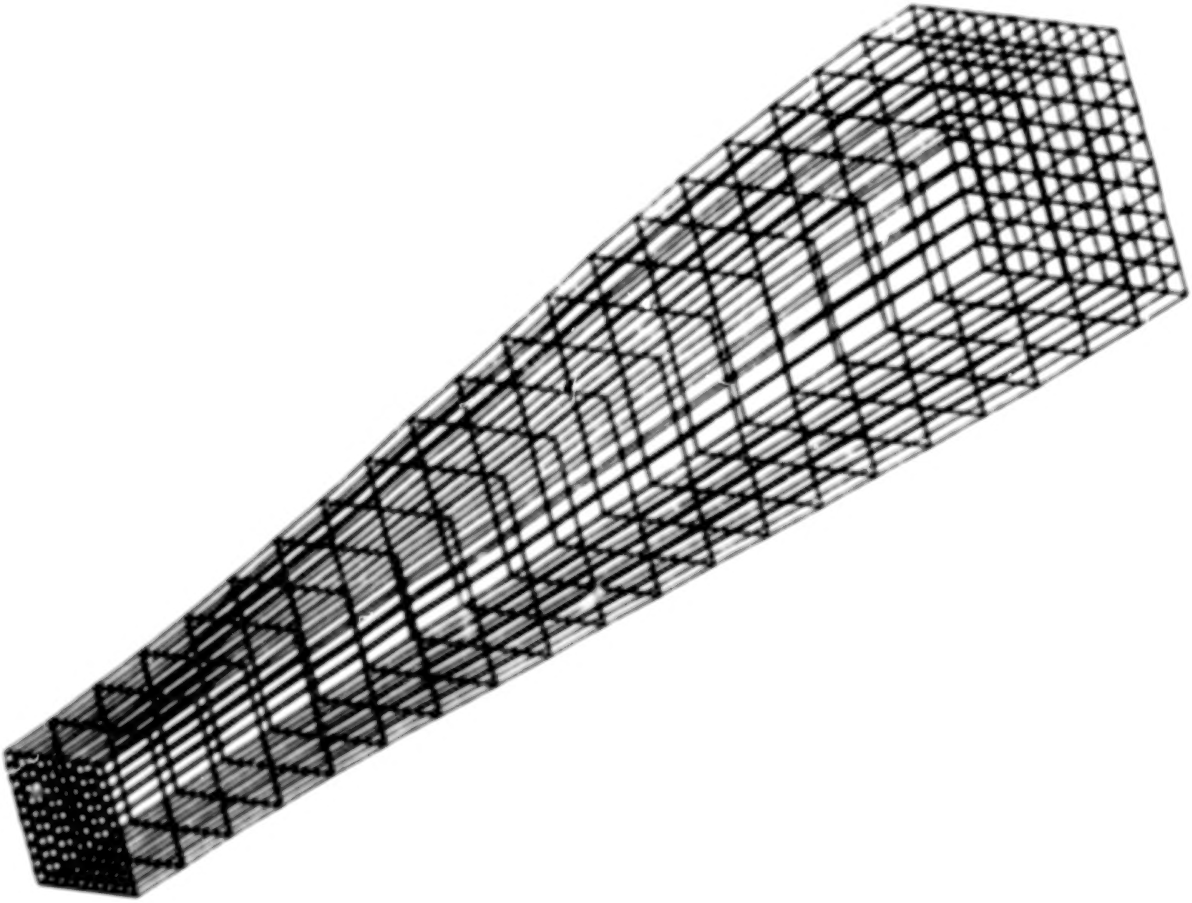
EXTERNAL FLOW GRID
(Non-Orthogonal Curvilinear Coordinates)

The nodal network for the external flow over an ogive cylinder illustrates the capability of the GIM code geometry package to stretch the nodal distribution. The grid is very compact in the leading edge region and greatly expanded in the far field areas. The axial points follow the body surface and could generally be called "body-oriented coordinates" in the nomenclature of the literature. The radial grid lines are not necessarily normal to the lateral lines or to the body surface. The GIM code, through its "nodal-analog" concept can operate on this general non-orthogonal curvilinear grid.



THREE-DIMENSIONAL GRID
(Simple Rectilinear Coordinates)

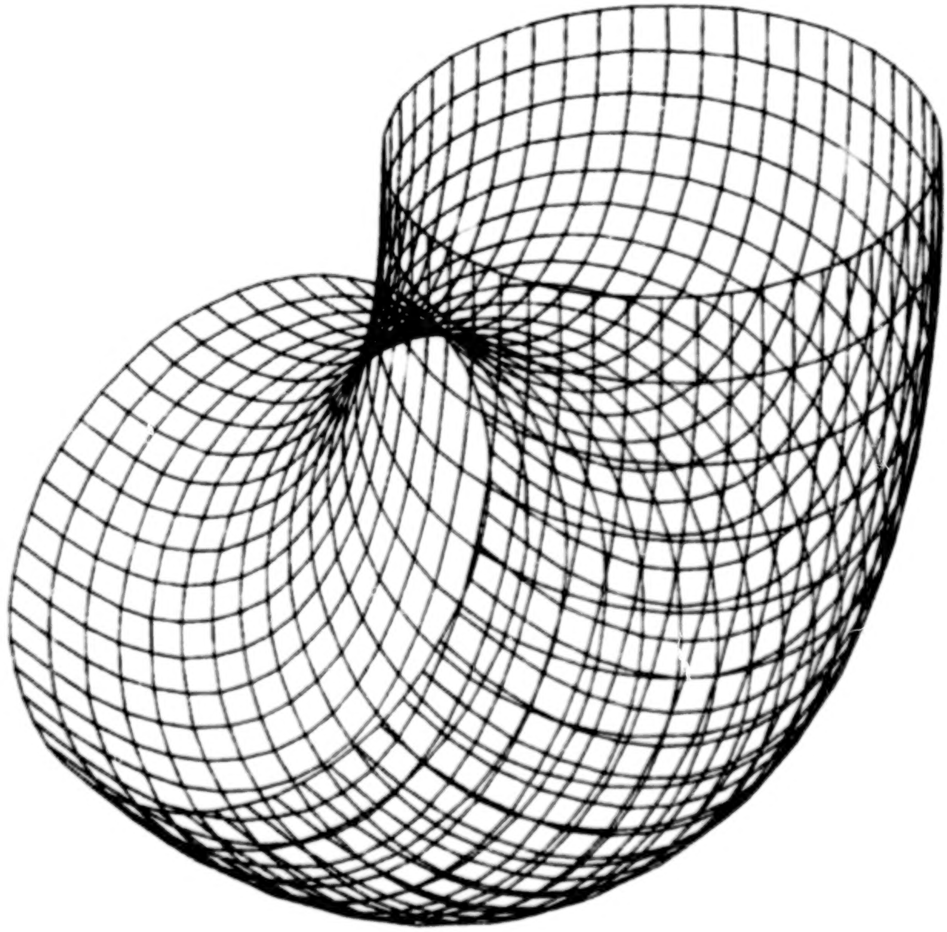
Supersonic flow in expanding ducts of arbitrary cross section is a common occurrence in computational fluid dynamics. This figure illustrates a simple grid for a three-dimensional duct whose cross section varies sinusoidally with the axial coordinate. The "top" wall and the "front" wall have this sinusoidal variation while the "bottom" and "back" walls are flat plates. The grid shown was used to resolve the expanding-recompressing supersonic flow including the intersection of the two shock sheets.



THREE-DIMENSIONAL GRID
(Pipe Flow in a 90 deg Elbow Turn)

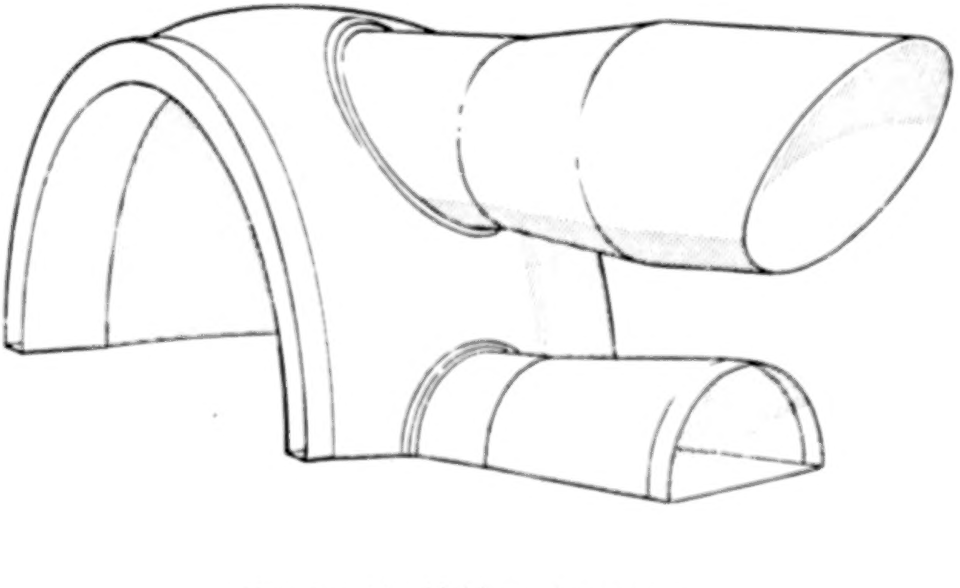
There are numerous flow fields of interest which contain a sharp turn inside a smooth pipe. The GIM code has treated certain of these for application to jet deflector nozzle flow in VTOL aircraft. The portion of a grid shown in the adjacent figure was used for this calculation.

The 90 deg elbow demonstrates the capability to model three-dimensional non-Cartesian geometries. The internal nodes were emitted for clarity. The elbow grid was generated by employing edge-of-revolution surfaces with circular arc segments as the edges being revolved.



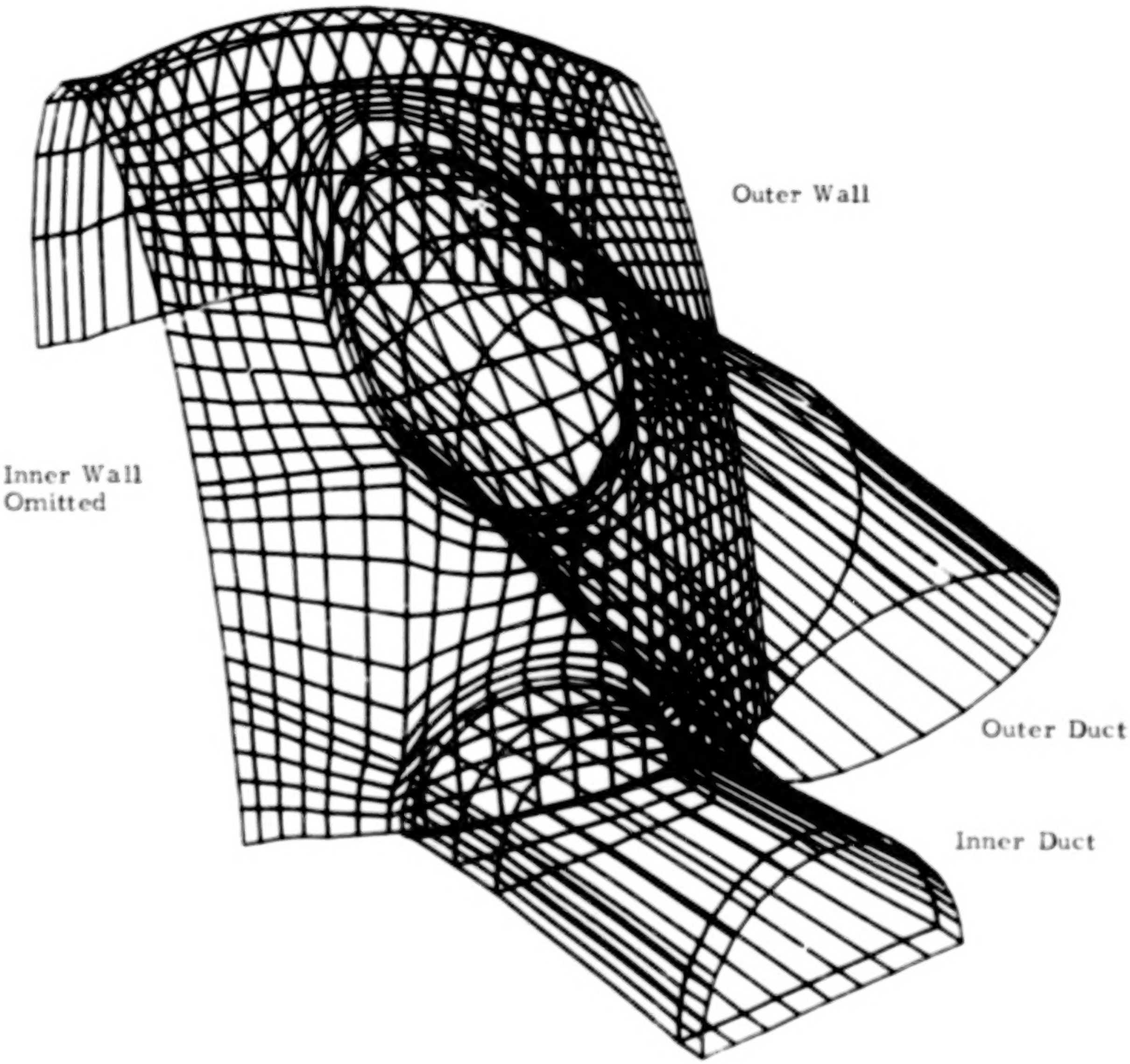
GRID FOR SPACE SHUTTLE MAIN ENGINE
(Hot Gas Manifold Geometry Model)

The recent problems encountered with the Space Shuttle main engine tests have resulted in a GIM code analysis of the system. The "hot gas manifold" is a portion of this analysis for the high pressure turbopump system. The grid shown in the adjacent figure was used for this calculation. Only a small number of nodes are shown for clarity; the full model consists of approximately 14,000 nodes. The extreme complexity of this geometry illustrates the necessity of using a GIM-like technique. Transforming this case to a square box computational domain is, of course, impossible. The results of the GIM code analysis agree qualitatively with flow tests that have been run on the hot gas manifold.



Hot Gas Manifold Configuration

GRID FOR SPACE SHUTTLE MAIN ENGINE
(Hot Gas Manifold Geometry Model)


Outer Wall
Inner Wall Omitted
Outer Duct
Inner Duct

SUMMARY

- Finite difference grids can be generated for very general configurations by using multivariate blending function interpolation.

- The GIM code difference scheme operates on general nonorthogonal curvilinear coordinate grids.

- This scheme does not require a single transformation of the flow domain onto a square box. Thus, GIM routines can indeed treat arbitrary three-dimensional shapes.

- Grids generated for both internal and external flows in two and three dimensions have shown the versatility of the algebraic approach.

- The GIM code integration module has successfully computed flows on these complex grids, including the Space Shuttle main engine turbopump system.

- Plans for future application of the code include supersonic flow over missiles at angle of attack and three-dimensional, viscous, reacting flows in advanced aircraft engines. Plans for future grid generation work include schemes for time-varying networks which adapt themselves to the dynamics of the flow.

BIBLIOGRAPHY

Spradley, L. W., J. F. Stalnaker and A. W. Ratliff, "Computation of Three-Dimensional Viscous Flows with the Navier-Stokes Equations," AIAA Paper 80-1348, July 1980.

Spradley, L. W., and M. L. Pearson, "GIM Code User's Manual for the STAR-100 Computer," NASA-CR-3157, Langley Research Center, Hampton, Va., 1979.

Spradley, L. W., P. G. Anderson and M. L. Pearson, "Computation of Three-Dimensional Nozzle-Exhaust Flows with the GIM Code," NASA CR-3042, Langley Research Center, Hampton, Va., August 1978.

Prozan, R. J., L. W. Spradley, P. G. Anderson and M. L. Pearson, "The General Interpolants Method," AIAA Paper 77-642, June 1977.

Gordon, W. J., and C. A. Hall, "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," J. Numer. Math., Vol. 1, 1973, pp. 461-477.

Component-Adaptive Grid Embedding

E. H. Atta
Lockheed-Georgia Company

Introduction:

One of the major problems related to transonic flow prediction about realistic aircraft configuration is the generation of a suitable grid which encompasses such configurations. In general, each aircraft component (wing, fuselage, nacelle) requires a grid system that is usually incompatible with the grid systems of the other components; thus, the implementation of finite-difference methods for such geometrically-complex configurations is a difficult task.

In this presentation a new approach is developed to treat such a problem. The basic idea is to generate different grid systems, each suited for a particular component. Thus, the flow field domain is divided into overlapping subdomains of different topology. These grid systems are then interfaced with each other in such a way that stability, convergence speed and accuracy are maintained.

Model:

To evaluate the feasibility of the present approach a two-dimensional model is considered (figure 1). The model consists of a single airfoil embedded in rectangular boundaries, representing an airfoil in a wind tunnel or in free air. The flow field domain is divided into two overlapping subdomains, each covering only a part of the whole field. The inner subdomain employs a surface-fitted curvilinear grid generated by an elliptic grid-generator (ref. 1), while the outer subdomain employs a cartesian grid. The overlap region between the two subdomains is bounded by the outer boundary of the curvilinear grid and the inner boundary of the cartesian grid.

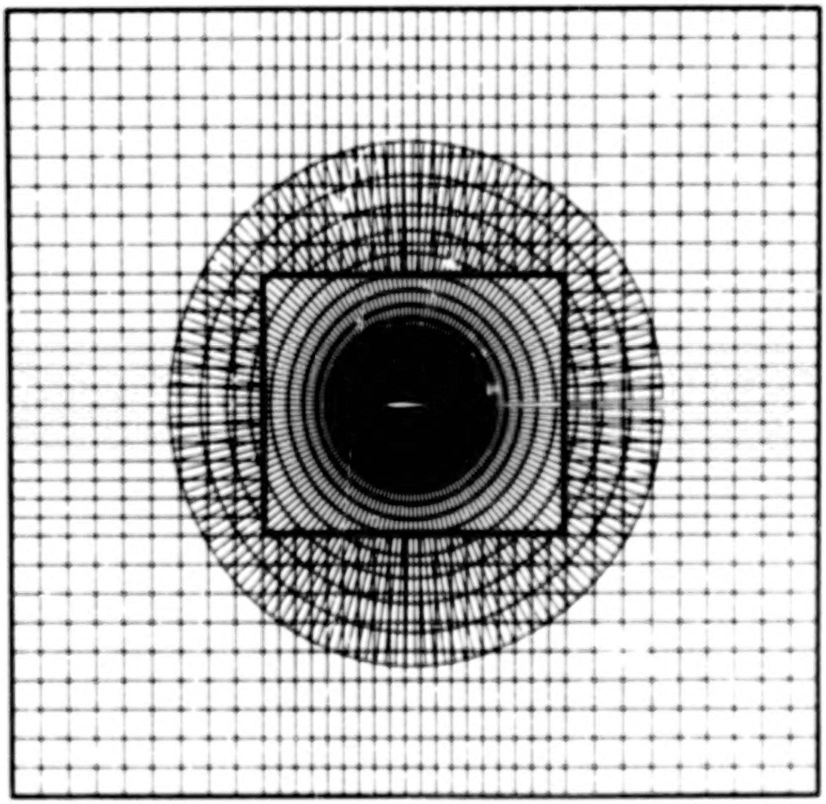


Figure 1.- Composite grid for an airfoil.

Approach:

Figure 2 shows the two subdomains (A,B) of the flow field; each has a grid adapted to suit its geometry. The flow in both subdomains is governed by the transonic full-potential equation. While a Neumann-type boundary condition is used at the inner boundary of subdomain B (overlap inner boundary), a Dirichlet-type boundary condition is used at the outer boundary of subdomain A (overlap outer boundary). These boundary conditions are updated during the solution process. The implicit approximate factorization scheme is used in both grid systems. The code of ref. 1 is modified to fit into the present scheme.

The solution process is performed in cycles, starting by solving for the flow field in subdomain A, then switching after a number of iterations to solve for the flow field in subdomain B. During each cycle the overlap boundary conditions are updated by using a two dimensional second order Lagrangian interpolation scheme. This process is then repeated until convergence is achieved in both subdomains.

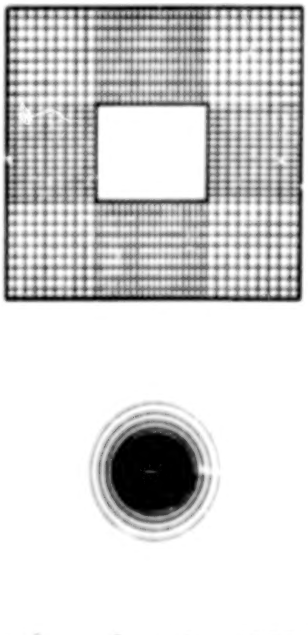


Figure 2.- Grid topology for the different subdomains.

Comparison with a homogeneous grid:

The results of the present method are compared with the results obtained from using one homogeneous grid for the entire flow field (ref. 1). In all the test cases considered, a standard grid with (31 x 147) points and a circular outer boundary located 6 chord-lengths away from the airfoil are used. (See figure 3.)

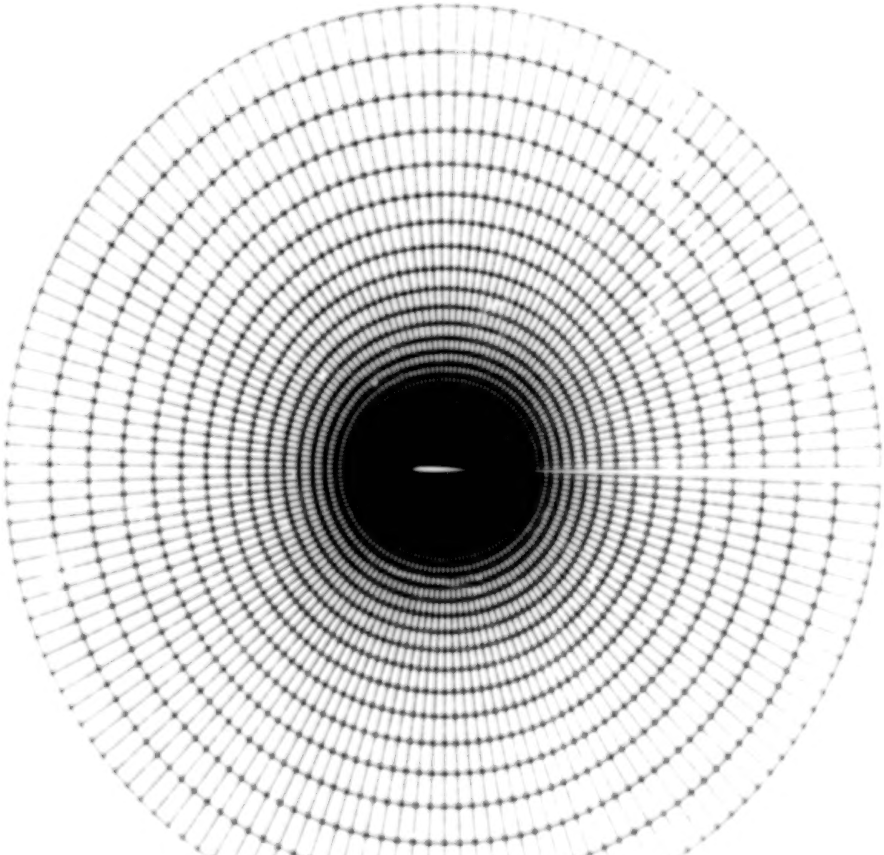


Figure 3.- Uniform grid for an airfoil (ref. 1).

Computed Results:

Results of the present method are compared with the results obtained from the code of ref. 1. Two sets of parameters affecting the performance of the numerical scheme are listed in tables I and II. Figures 4 and 5 display the pressure-coefficient distributions for a NASA-0012 airfoil resulting from the flow field solutions. The results are in good agreement for both subcritical and supercritical cases; savings in computing time are achieved by reducing the size of the flow field covered by the curvilinear grid (subdomain A).

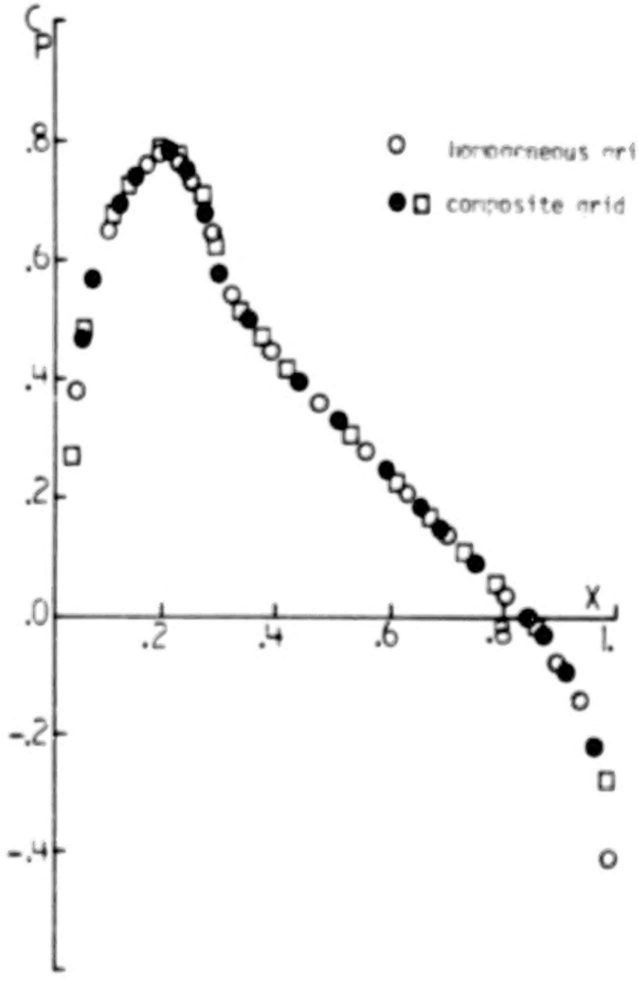


Figure 4.- Comparison of pressure coefficient for NACA-0012. (M_∞ = 0.75, α = 0.)

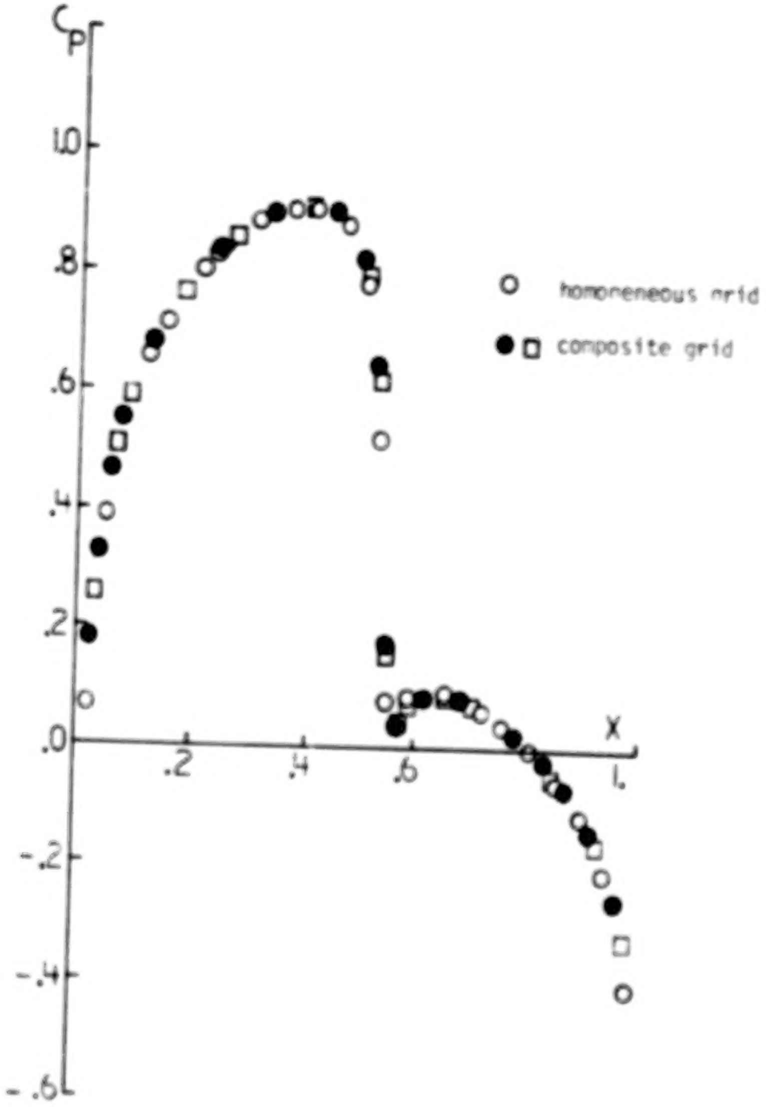


Figure 5.- Comparison of pressure coefficient for NACA-0012. (M_∞ = 0.8, α = 0.)

Flow Field Topology :

The extent of the overlap region between the different grids and the relative size of each subdomain are the main factors affecting the accuracy and convergence speed of the present scheme. Figure 6 shows the flow field topology for several test cases. In these cases the overlap extent and subdomain sizes are varied to determine their optimum values that will minimize the computing effort, while maintaining a reasonable accuracy.

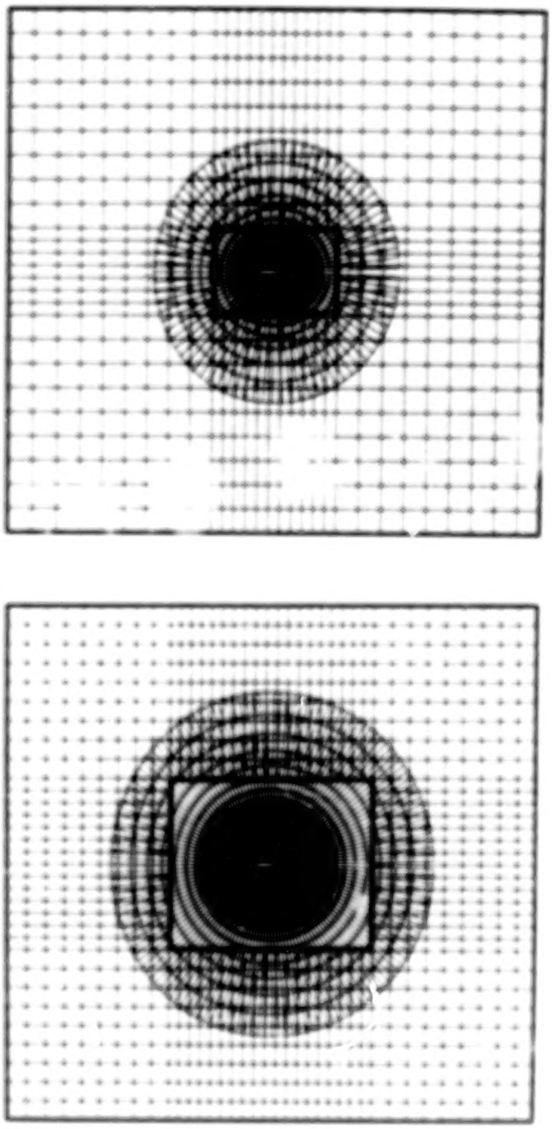


Figure 6.- Flow-field topology with different grid-overlap.

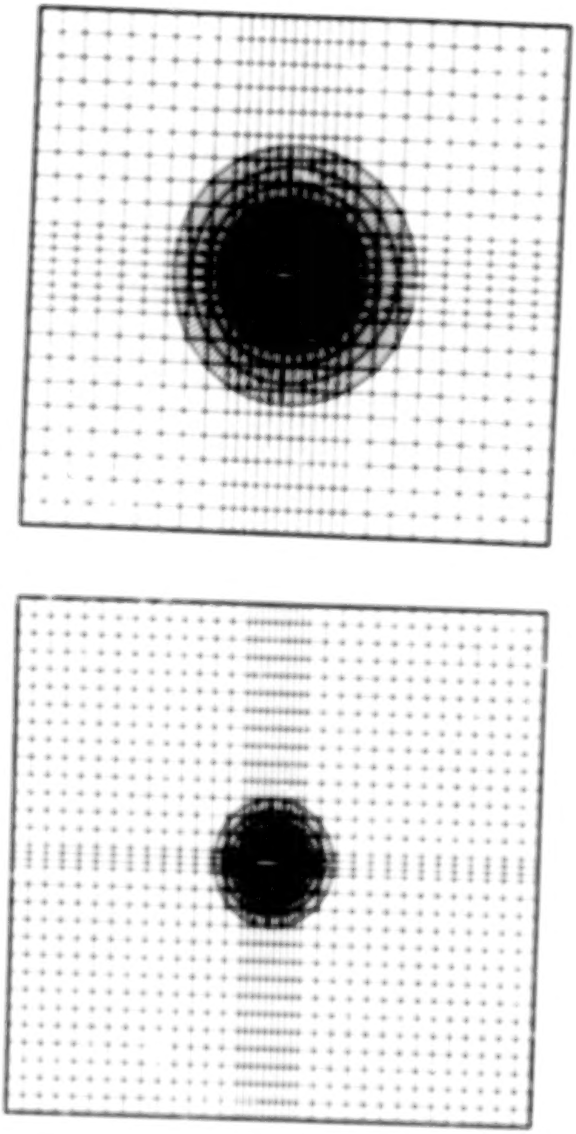


Figure 6.- Concluded.

Overlap arrangement:

Test cases with different grids-arrangement are compared to determine the optimum choice for the extent of the overlap region. A work factor w [number of iterations for convergence x number of grid points (curvilinear grid)] is taken as a measure of the computing effort. Numerical results show that increasing the extent of the overlap region decreases the number of iterations for convergence; however, this also increases the computing effort (figure 7). To minimize the computing time the Cartesian grid should overlap 15-25% of the curvilinear grid, and the inner boundary of the Cartesian grid should not be located less than 0.25 chord-length away from the airfoil.

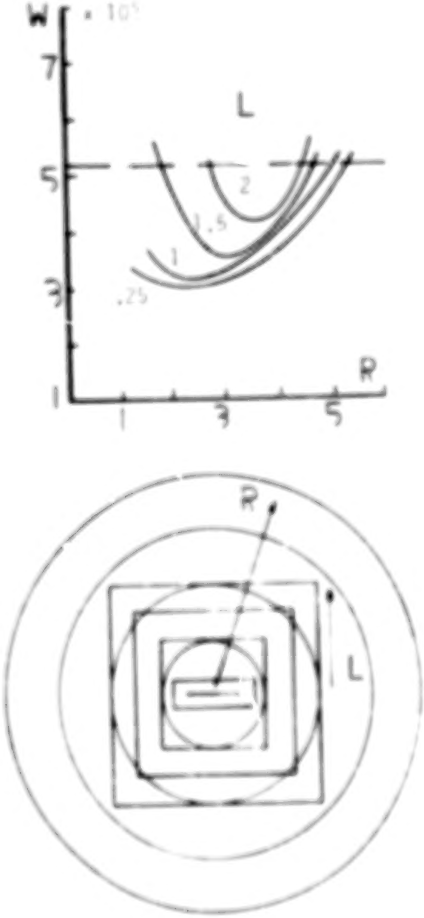


Figure 7.- Effect of overlap parameters on work factor w. (NACA-0012, $M_\infty = 0.8$, $\alpha = 0$.)

Computed Results:

The use of nonoptimal parameters for grids arrangement (overlap extent, relative grid sizes) can produce inaccurate results and/or slow down convergence. The Peaky pressure coefficient distribution shown in figure 8 is corrected by increasing the extent of the overlap region described in Table III.

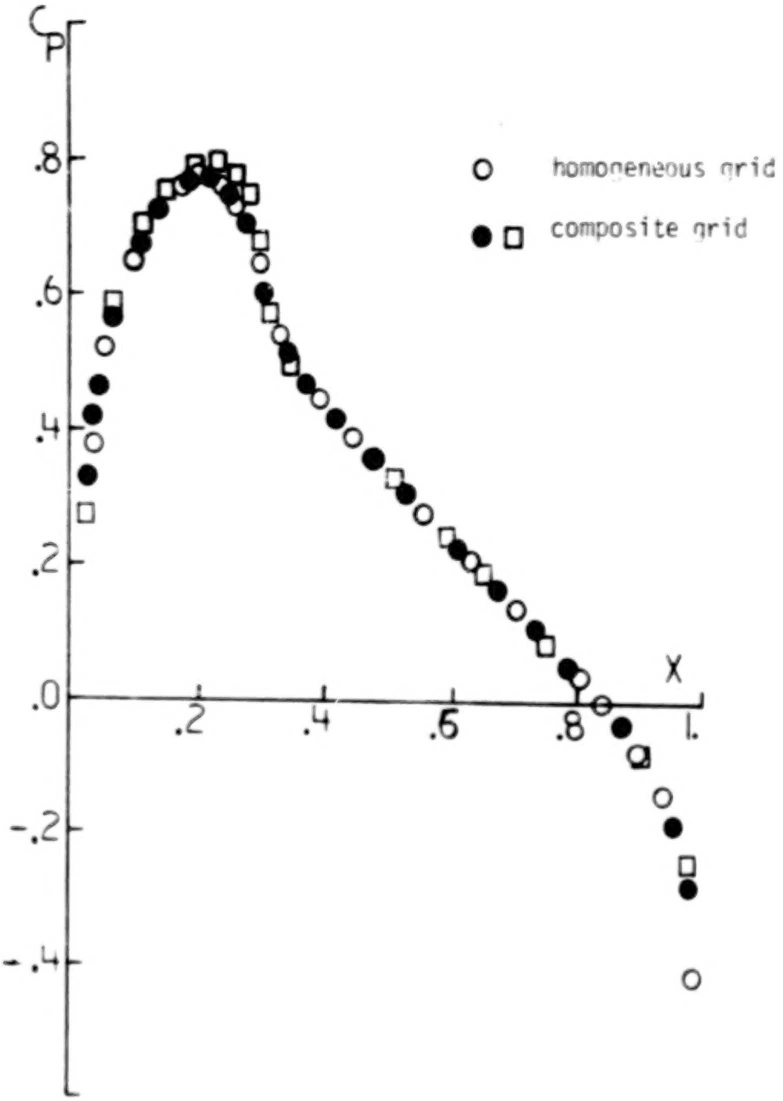


Figure 8.- Comparison of pressure coefficient for NACA-0012. (M_∞ = 0.75, α = 0.)

Computed Results:

Figures 9 and 10 display the pressure coefficient distributions for two lifting cases for the parameters described in Tables IV and V. The evolution of circulation, and hence lift, is slowed down as the solution process alternates between the different grids. This is dealt with by decreasing the number of iterations performed in each grid.

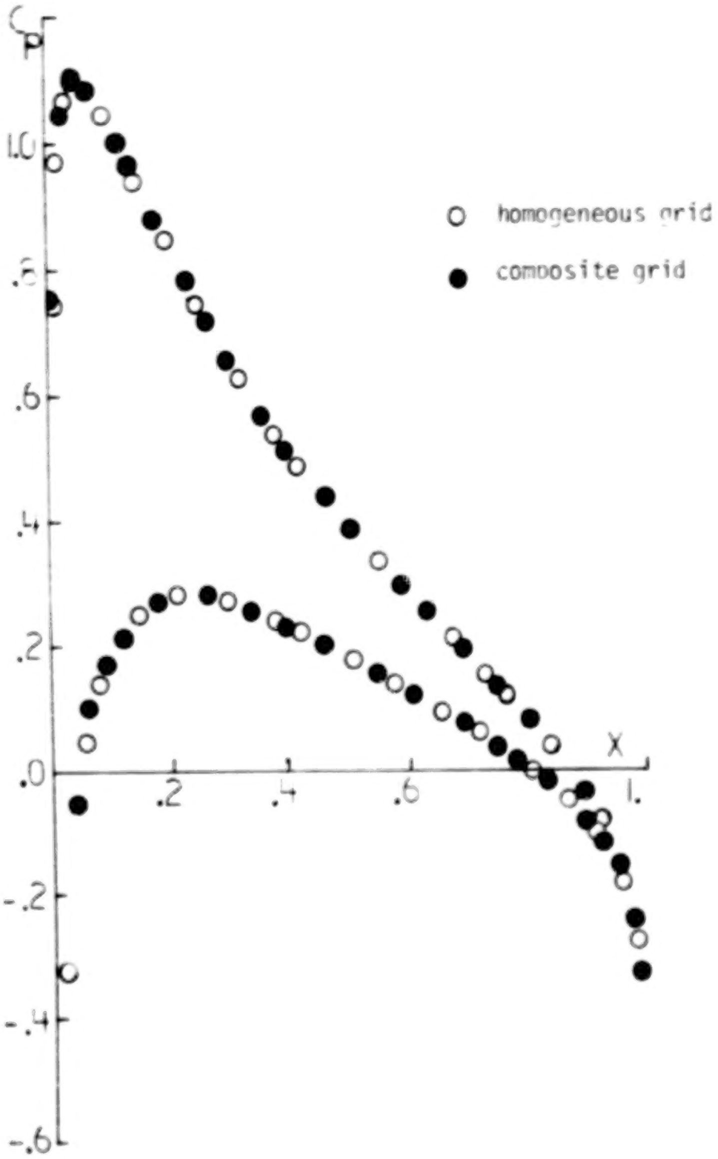


Figure 9.- Comparison of pressure coefficient for NACA-0012. (M_∞ = 0.63, α = 2°.)

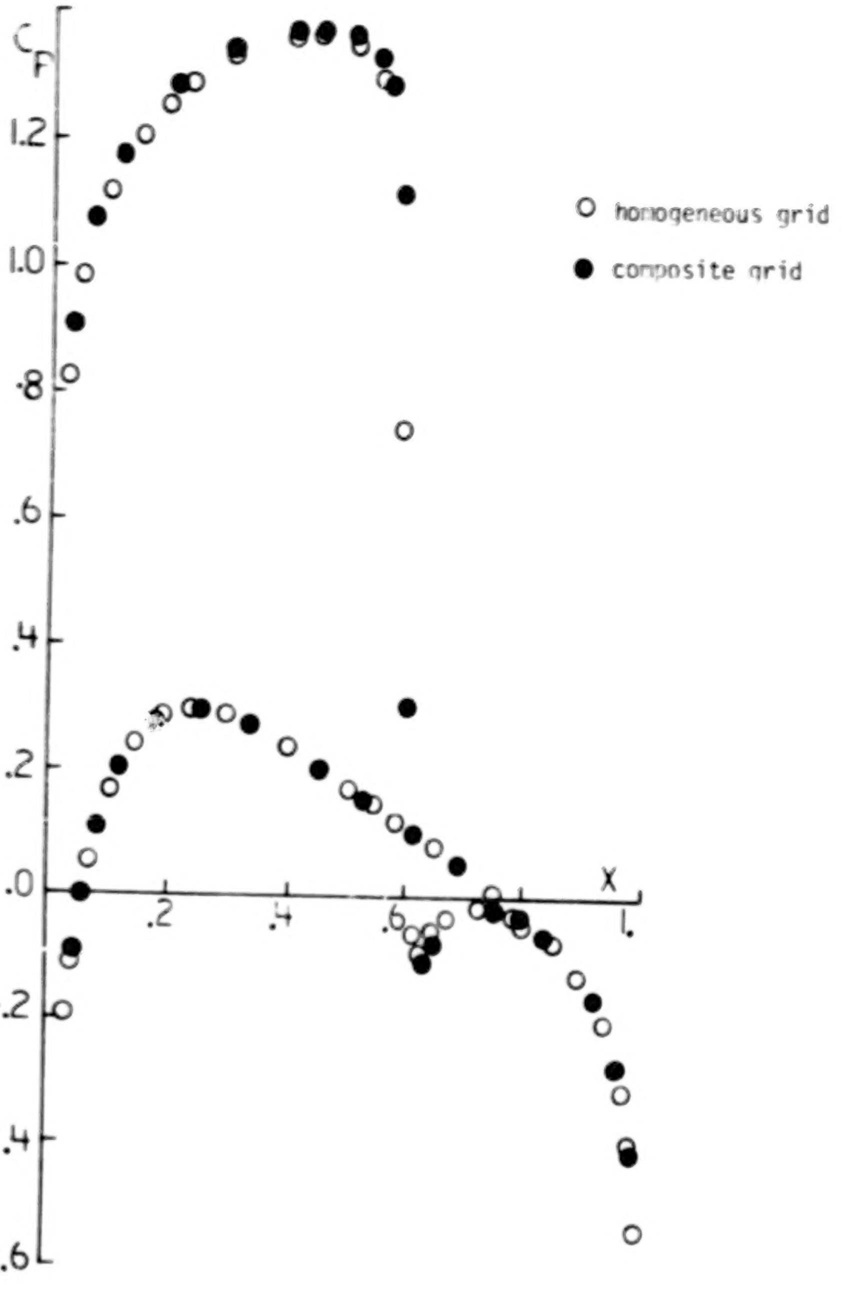


Figure 10.- Comparison of pressure coefficient for NACA-0012.
(M_∞ = 0.75, α = 2°.)

Errors in sonic line position:

Should the shock wave extend into the overlap region, the interpolation process can produce errors in the shock location and strength. Comparisons of the results of the present method with those of a homogeneous grid shows that the maximum relative error did not exceed 1.5%. (See figure 11.)

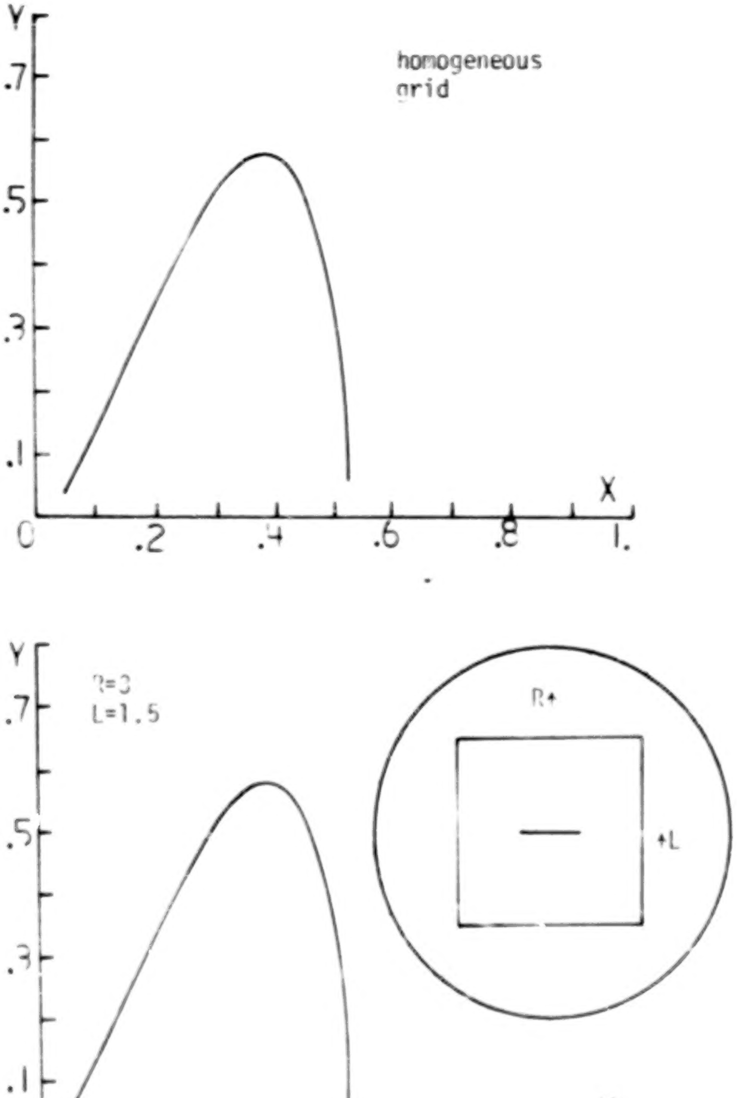


Figure 11.- Effect of interface location on sonic line position. (NACA-0012, $M_\infty = 0.8$, $\alpha = 0$.)

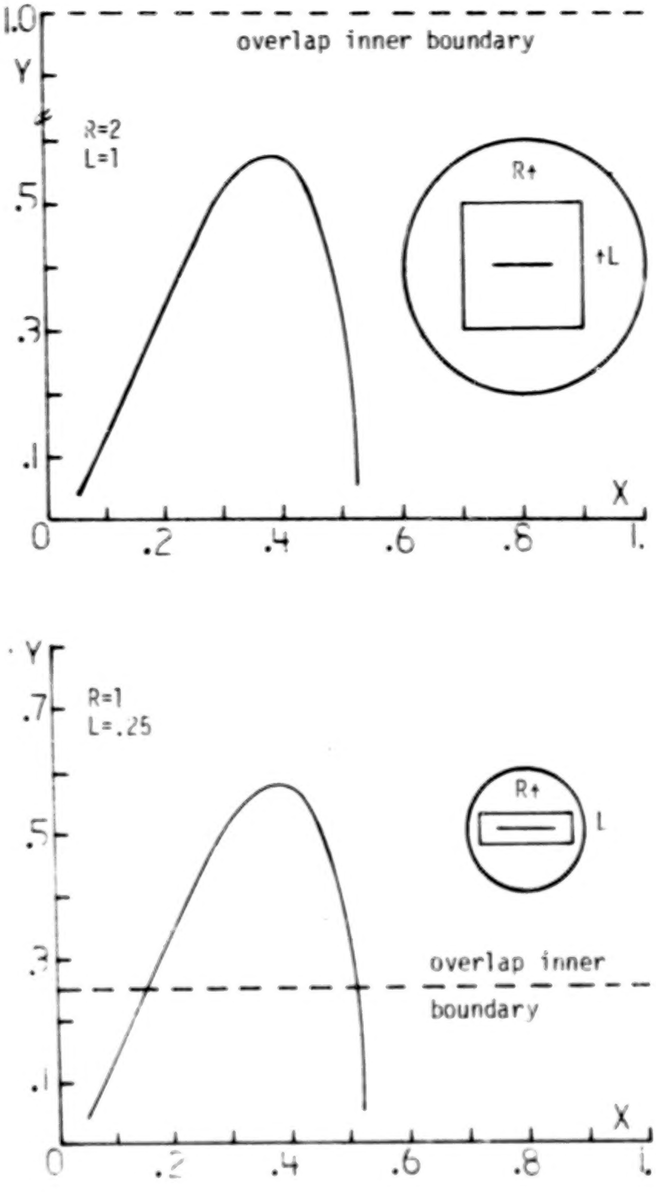


Figure 11.- Concluded.

Conclusion:

A method for interfacing grid systems of different topology is developed. This offers a new approach to the problem of transonic flow prediction about multiple-component configurations. The method is implemented in a 2-D domain containing two grid systems of different topology. The numerical scheme in the present method proved to be stable and accurate. Savings in computer time and/or storage is achieved by the proper choice of the overlap region between the different grids.

Reference:

1. Holst, T. L., "Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh," AIAA J., Vol. 17, No. 10, October 1979.

TABLE I

	Code of Ref. 1 TAIR Code	Present Method	
Curvilinear grid	31 x 147	15 x 147	21 x 147
Cartesian grid		30 x 30	30 x 30
% cpu time reduction as compared to TAIR Code		30%	10%
location of subdomain B outer boundary		6 chord-length	6 chord-length
Location of subdomain B inner boundary		1 chord-length	2 chord-length
Location of subdomain A outer boundary		1 chord-length	4 chord-length
number of cycles for convergence		9	10

TABLE II

	Code of Ref. 1 TAIR Code	Present Method	
Curvilinear grid	31 x 147	18 x 147	14 x 147
Cartesian grid		30 x 30	50 x 50
% cpu time reduction as compared to TAIR Code		20%	10%
location of subdomain B outer boundary		6 chord-length	6 chord-length
location of subdomain B inner boundary		1 chord-length	1/4 chord-length
location in subdomain A outer boundary		2 chord length	1 chord length
number of cycles for convergence		12	15

TABLE III

	Code of Ref. 1 TAIR Code	Present Method	
Curvilinear grid	31 x 147	10 x 147	15 x 147
Cartesian grid		30 x 30	40 x 40
location of subdomain B outer boundary		6 chord-length	6 chord-length
location of subdomain B inner boundary		1/4 chord-length	1/4 chord-length
location of subdomain A outer boundary		1.5 chord-length	1 chord-length

TABLE IV

	Code of Ref. 1 TAIR Code	Present Method
Curvilinear grid	31 x 147	15 x 147
Cartesian grid		30 x 40
% cpu time reduction as compared to TAIR Code		39%
location of subdomain B outer boundary		6 chord-length
location of subdomain B inner boundary		1 chord-length
location of subdomain A outer boundary		3 chord-length
lift coefficient	0.334	0.337
number of cycles for convergence		16

TABLE V

	Code of Ref. 1 TAIR Code	Present Method
Curvilinear grid	31 x 147	21 x 147
Cartesian Grid		30 [illegible]
% cpu time reduction as compared to TAIR Code		2%
location of subdomain B outer boundary		6 chord-length
location of subdomain B inner boundary		2 chord-length
location of subdomain A outer boundary		4 chord-length
lift coefficient	0.574	.584
number of cycles for convergence		14

GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS*

A. J. BAKER & P. D. MANHARDT

UNIVERSITY OF TENNESSEE/KNOXVILLE,
AND COMCO, INC, KNOXVILLE, TN

ABSTRACT

The generation of metric coefficients of the coordinate transformation from a generally curved-sided domain boundary to the unit square (cube) is required for efficient solution algorithms in computational fluid mechanics. An algebraic procedure is presented for establishment of these data on the union of arbitrarily selected sub-domains of the global solution domain. A uniformly smooth progression of grid refinement is readily generated, including multiple specification of refined grids for a given macro-element domain discretization. The procedure is illustrated as generally applicable to non-simply connected domains in two- and three-dimensions.

*Research principally supported by USAF Grant No. AFOSR-79-0005.

COMPUTATIONAL REQUIREMENT

NAVIER-STOKES EQUATIONS

$$L(q_i) = \frac{\partial q_i}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j q_i + f_{ij}\right] = 0$$

$$\ell(q_i) = a_1 q_i + a_2 \frac{\partial q_i}{\partial x_j} \hat{n}_j + a_3 = 0$$

COORDINATE TRANSFORMATION

$$x_i = x_i(\eta_j) \qquad \frac{\partial}{\partial x_j} = \frac{\partial}{\partial \eta_k}\left[\quad\right]\frac{\partial \eta_k}{\partial x_j}$$

$$J^{-1} = \left[\frac{\partial \eta_k}{\partial x_j}\right] \qquad \bar{u}_k = \frac{\partial \eta_k}{\partial x_j} u_j$$

NUMERICAL SOLUTION ALGORITHM

$$S_e\left[\{DET\underline{J}\}_e^T[M3000]\{QI\}_e^{\prime} - \{UBAR\underline{K}\}_e^T[M30\underline{K}0]\{QI\}_e - \{ETA\underline{KL}\}_e^T[M30\underline{K}0]\{F\underline{L}I\}_e\right] = \{0\}$$

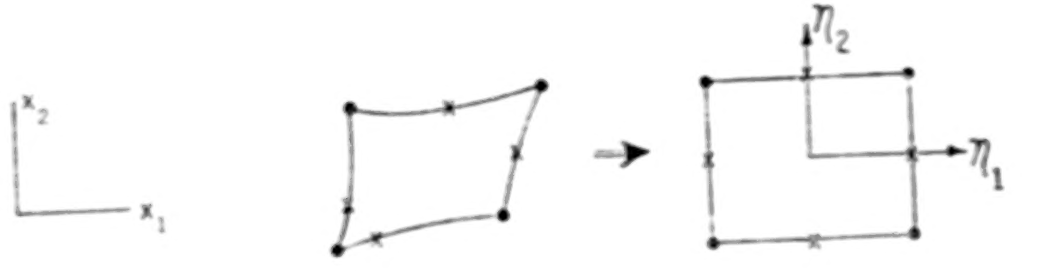
DISCUSSION

The Navier-Stokes equations contain the vector divergence operator. The required transformation projects x_i onto η_j with coordinate surfaces defined coincident with solution domain boundaries. The Cartesian description of dependent variables is retained, while the convection velocity is expressed in contravariant scalar components. The numerical solution implementation requires nodal distributions of components of the forward and inverse Jacobians, and $\underline{J}$, $\underline{K}$, and $\underline{L}$ are tensor summation indices.

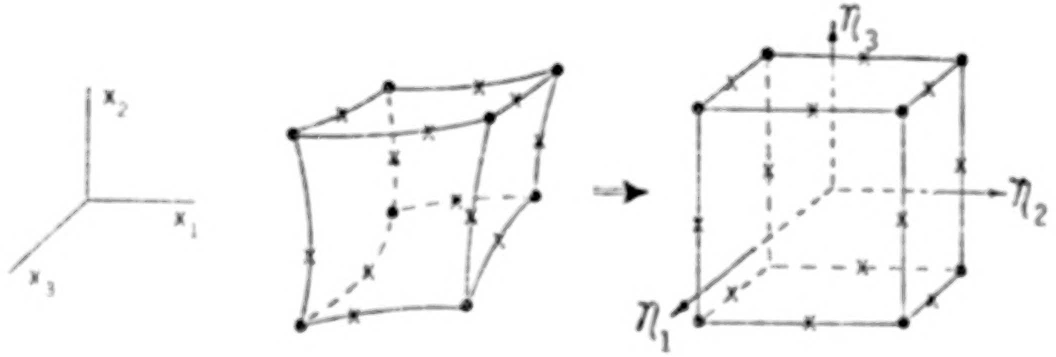
LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATION

PHYSICAL DOMAIN **TRANSFORMED DOMAIN**

$$x_i \equiv \{N_2(\vec{\eta})\}^T\{XI\}_e$$



Two-Dimensional

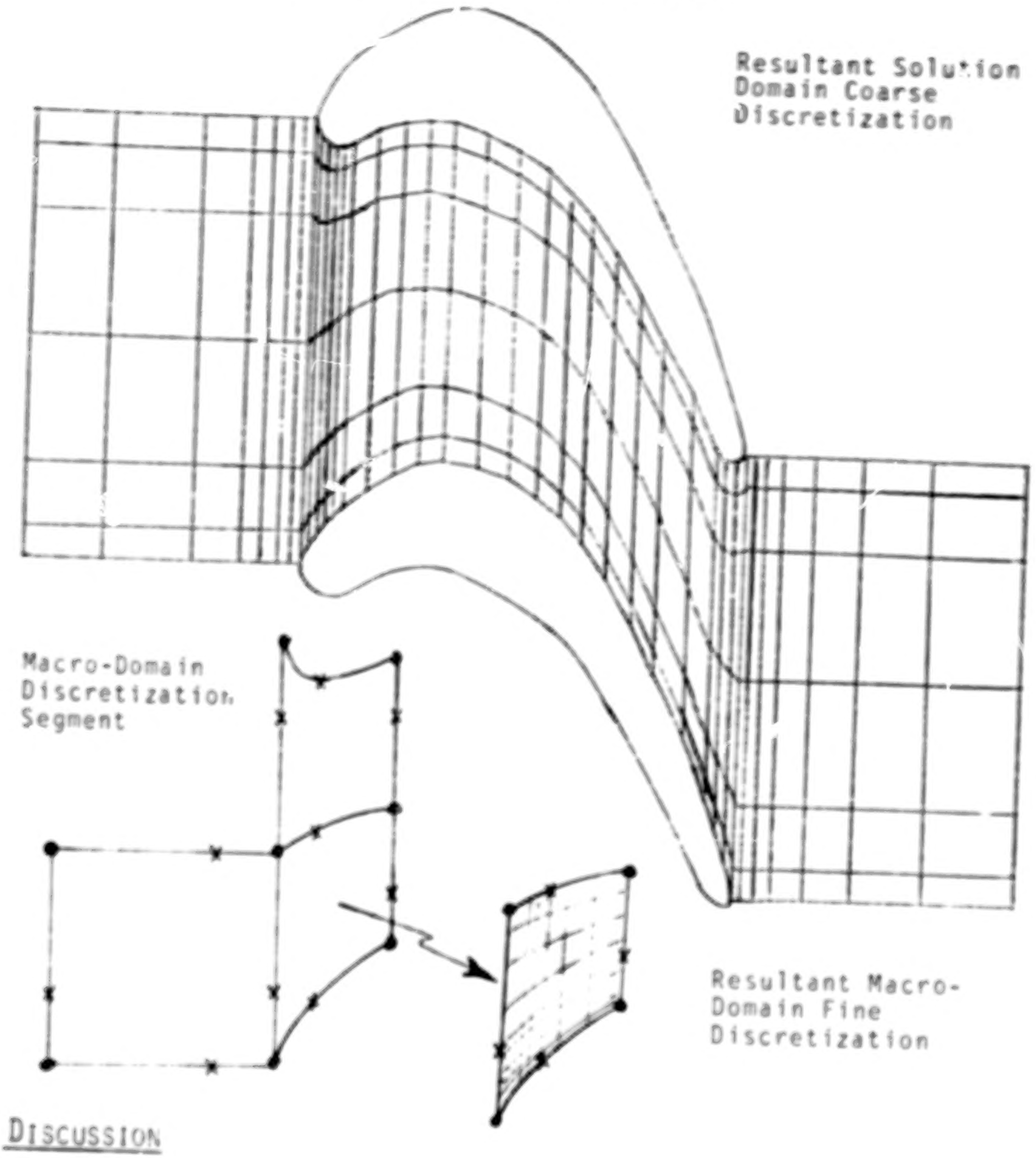


Three-Dimensional

DISCUSSION

The bi-quadratic cardinal basis $\{N_2(\vec{\eta})\}$ transforms the vertex and non-vertex node coordinate description of a smooth region of R^n onto the unit square or cube spanned by the locally rectangular Cartesian coordinate system $\vec{\eta}$. The inverse transformation J^{-1} is non-singular for a range of non-midpoint definitions of the non-vertex node coordinates $\{x\}$, yielding a non-uniform discretization on R^n.

EXAMPLE: COMPRESSOR BLADE ROW


Resultant Solution
Domain Coarse
Discretization
Macro-Domain
Discretization
Segment
Resultant Macro-
Domain Fine
Discretization

DISCUSSION

Three of the ten macro-domains, used to form the blade row discretization, are shown. The non-midside location of non-vertex nodes produces the non-uniform grid, only a few gridlines of which are shown. The inset illustrates a fine discretization of one macro-domain. The coordinates of all nodes on boundaries of macro-domains are unique.

DETAILS OF THE COORDINATE TRANSFORMATION

NODAL COORDINATES $\{XI\}$:

$$x_i = \{N_i(\eta_j)\}^T\{XI\}_e$$

WHERE:

$$\{N_i(\eta_j)\} = \frac{1}{4}\left\{\begin{array}{l} (1 - \eta_1)(1 - \eta_2)(-\eta_1 - \eta_2 - 1) \\ (1 + \eta_1)(1 - \eta_2)(\eta_1 - \eta_2 - 1) \\ (1 + \eta_1)(1 + \eta_2)(\eta_1 + \eta_2 - 1) \\ (1 - \eta_1)(1 + \eta_2)(-\eta_1 + \eta_2 - 1) \\ 2(1 - \eta_1^2)(1 - \eta_2) \\ 2(1 + \eta_1)(1 - \eta_2^2) \\ 2(1 - \eta_1^2)(1 + \eta_2) \\ 2(1 - \eta_1)(1 + \eta_2^2) \end{array}\right\}$$

JACOBIANS

$$J \equiv \left[\frac{\partial x_i}{\partial \eta_j}\right] = J(\eta_j, XI)$$

$$J^{-1} \equiv \left[\frac{\partial \eta_j}{\partial x_i}\right] = \frac{1}{\det J}\,[\text{cofactors of } J]^T$$

$$= J^{-1}(\eta_j, XI)$$

DISCUSSION

Within a macro-domain, the components of both J and J^{-1} are continuous functions of η_j and the global macro-node coordinate pairs (triples) $\{XI\}$, $1 \leq I \leq n$. Each global coordinate x_i possesses an independent transformation; the corresponding Jacobian must be of rank n to assure existence of J^{-1}. Once the matrix elements of $\{XI\}$ are defined, selection of any coordinate (η_1, η_2) defines a unique coordinate pair (x_1, x_2), i.e., a mesh point on the refined grid in physical space.

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B1
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

BLANK PAGE

BLANK PAGE

Grid Generation for Time Dependent Problems: Criteria and Methods

Marsha Berger, William Gropp and Joseph Oliger
Department of Computer Science
Stanford University

Abstract

We consider the problem of generating local mesh refinements when solving time dependent partial differential equations. We first discuss the problem of creating an appropriate grid, given a mesh function h defined over the spatial domain. A data structure which permits efficient use of the resulting grid is described. Secondly, we show that a good choice for h is an estimate of the local truncation error, and we discuss several ways to estimate it. We conclude by comparing the efficiency and implementation problems of these error estimates.

WHAT ADAPTIVE MESH GENERATION FOR TIME DEPENDENT PDE'S

OBJECTIVES REDUCE # MESH PTS
MINIMIZE OVERHEAD

TRADEOFF: EXTRA PTS. VS. EXTRA LOGIC

REQUIREMENTS

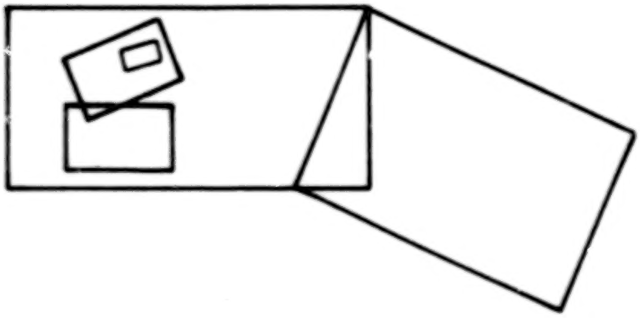
- MARCHING ALGORITHMS WILL BE USED
- COMPUTING TRANSIENT SOLN BY FINITE DIFF.
- TIMESTEP SMALLER ON FINER GRIDS, MESH RATIO CONSTANT
- GRIDS MUST CHANGE WITH TIME
- COARSEST GRID DOES NOT CHANGE WITH TIME

DESCRIPTION OF GRIDS

- LOCALLY UNIFORM
- RECTANGLES OF ARBITRARY ORIENTATION. EXTENSIONS TO CURVILINEAR GRIDS FITS INTO SAME FRAMEWORK
- SUPPOSE BASE GRID $G_0 = \bigvee_j G_{0,j}$ FORM HIERARCHY OF

  NESTED GRIDS WHERE EACH REFINED GRID IS WHOLLY CONTAINED IN A SINGLE COARSER GRID

$$G_l = \bigvee_j G_{l,j}$$

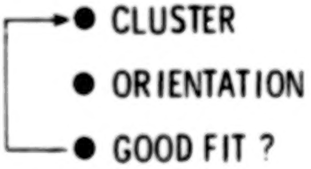


- REFINED GRIDS CAN BE CONSTRUCTED AUTOMATICALLY AT $t = 0$ FROM INITIAL DATA.

HOW GRIDS ARE FORMED

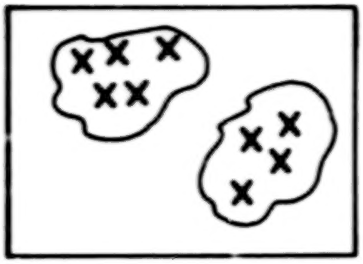
GIVEN A "MESH FUNCTION" $h(s, y)$ USED TO DETERMINE WHERE TO PLACE REFINED GRIDS.

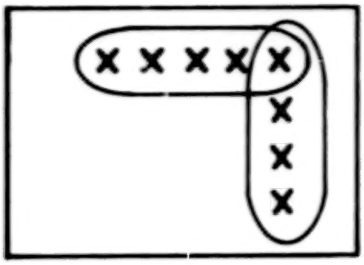
FLAG GRID PTS. WHERE $h(x, y) > \varepsilon$.



- CLUSTER
- ORIENTATION
- GOOD FIT ?

CLUSTERING





- NEAREST NEIGHBOR

  $d(\text{PT.}, \text{CLUSTER}) < d\text{max} \Longrightarrow$
  $\text{PT.} \in \text{CLUSTER}$

- SPANNING TREES

  CONNECT ALL PTS. ACCORDING TO SOME CRITERIA.
  BREAK LONGEST LINKS.

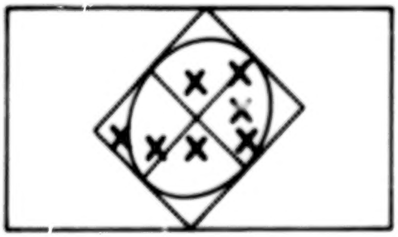
  - MINIMAL SPANNING TREES
  - MINIMUM DIAMETER TREES

ORIENTATION

- FIT ELLIPSE TO FLAGGED PTS. OF A CLUSTER USING 1ST AND 2ND MOMENTS.
- USE MAJOR AND MINOR AXES OF THE ELLIPSE TO GET RECTANGLE ORIENTATION.

(REF: D. GENNERY, "OBJECT DETECTION AND MEASUREMENT USING STEREO VISION") PROCS. IJCAI, 1979, pp 320-327

- FIT MIN. BOX TO INCLUDE FLAGGED PTS. + SMALL BUFFER ZONE FOR SAFETY.



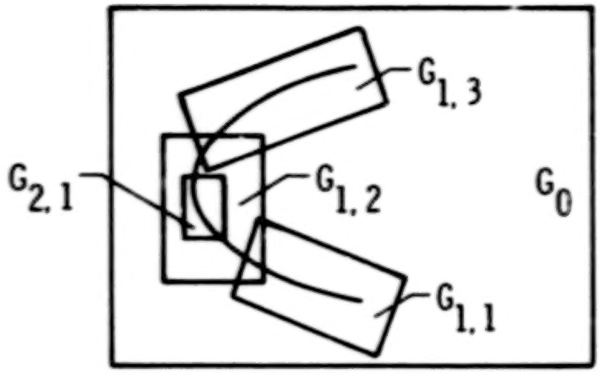
GOODNESS OF FIT

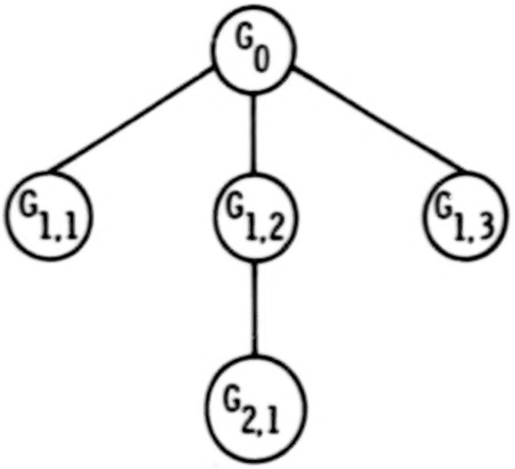
- RATIO OF FLAGGED TO UNFLAGGED PTS.
- IF TOO LOW, RECLUSTER AND REFIT.

KEEPING TRACK OF GRIDS

NESTING SUGGESTS USE OF TREE STRUCTURE

(REF: KNVTH, "ART OF COMPUTER PROGRAMMING", VOL. 1)





INFORMATION FOR EACH GRID

1) GRID LOCATION
2) SPATIAL AND TEMPORAL STEP SIZES
3) SIZE OF GRID
4) 3 TREE LINKS
5) PTR. TO INTERSECTING GRIDS
6) MAIN STORAGE AREA PTR.

POINTS TO NOTE

1) EASY TO HANDLE FAIRLY GENERAL REGIONS.

   ALL THE WORK IN SETTING UP THE PROBLEM IS IN SPECIFYING THE LOCATION OF THE COARSE GRID AND ITS CONSTITUENT RECTANGLES. THE REST IS AUTOMATIC.

2) EASY TO USE DIFFERENT METHODS ON DIFFERENT GRIDS.

WHAT IS h(x,y)?

WOULD LIKE TO EQUIDISTRIBUTE THE GLOBAL ERROR.

1D LINEAR THEORY SAYS IF CONTROL

(1) INITIAL ERROR
(2) BOUNDARY ERROR
(3) LOCAL TRUNCATION ERROR

AND METHOD IS STABLE FOR IBVP THEN THE METHOD CONVERGES.

(1) AND (2) CONTROLLED BY STD. MEANS

(3) CONTROLLED BY REFINING MESHES

USE LOCAL TRUNCATION ERROR FOR h (x, y).

REQUIREMENTS FOR LOCAL ERROR ESTIMATOR

- ACCURATELY MIMIC ERROR BEHAVIOR
- REASONABLY ACCURATE ESTIMATE - NOT NEC. A BOUND
- CHEAP TO COMPUTE
  FLEXIBLE - EASY TO SWITCH INTEGRATORS
- THE FEWER TIME LEVELS THE BETTER.

POSSIBLE ESTIMATORS

DIRECT ESTIMATION OF TRUNC. ERROR

- FIND LEADING TERM

(e.g. $\frac{k^2}{6} V_{ttt} + \frac{h^2}{6} V_{xxx}$)

- ESTIMATE BY DIVIDED DIFFERENCES

PROBLEMS

- HARD TO FIND LEADING TERM
- HARD TO CHANGE INTEGRATORS
- NO CHEAPER THAN OTHER ESTIMATES

LOWER ORDER ESTIMATES

(V_t, V_{tt})

- ESTIMATE SOLN. GROWTH IN TIME
- PROS - CHEAP, BETTER THAN GRADIENT ESTIMATES
- CONS - ACCURATE TRENDS BUT INACCURATE ESTIMATE OF MAGNITUDE.

GRADIENTS

- USE U_x

PROBLEMS

- EASY TO FOLL (e.g. FORCING FN.)
- NO CHEAPER THAN V_t
- GOOD ONLY FOR SHOCKS

DEFERRED CORRECTION

- USES 2 METHODS
- COMPUTE ERROR ESTIMATE AS A FUNCTION OF THE 2 SOLUTIONS

PROS	CONS
MOST ACCURATE ESTIMATES TESTED	EXTRA TIME LEVELS FOR 2ND METHOD
	DIFFICULT TO FIND 2ND METHOD AND ERROR RELATION

SPECIAL CASE (2h, 2k)

- 2ND METHOD USES SAME INTEGRATOR WITH DOUBLE THE STEP SIZES
- ERROR $\dfrac{V_{h,k} - \hat{V}_{2h,2k}}{2^{P+1} - 1}$

USE OF DIFFERENTIAL EQ. TO ELIMINATE TIME DERIV.

- USE $U_t = f(u, x, t)_x$ TO REPLACE TIME DERIVS. IN TRUNCATION ERROR

PROBLEMS

- MESSY TO FIND V_{ttt}
- VERY PROBLEM AND METHOD DEPENDENT
- USEFUL ONLY IF EXTREME PENALTY FOR USING EXTRA TIME LEVELS.

CONCLUSION

AUTOMATIC REFINED GRID GENERATION

- ARBITRARY ORIENTATION OF RECTANGLES
- LOW OVERHEAD OF GRID REPRESENTATION
- REFINEMENTS BASED ON (2h, 2k) ESTIMATES OF LOCAL TRUNCATION ERROR

GENERATIONS OF ORTHOGONAL SURFACE COORDINATES*

F. G. Blottner and J. B. Moreno
Sandia National Laboratories†
Albuquerque, NM 87185

An orthogonal surface-oriented coordinate system has been developed for three-dimensional flows where the computational domain normal to the surface is small. With this restriction the coordinate system requires orthogonality only at the body surface. The coordinate system is as follows: one coordinate measures distance normal to the surface while the other two coordinates are defined by an orthogonal mesh on the surface. One coordinate is formed by the intersection of the body surface and the meridional planes as illustrated in Figure 1 and gives the ϕ = constant lines. The other coordinate ξ, which is nondimensionalized with a characteristic length of the body geometry, measures the distance along the body surface when $\phi = 0$. This coordinate system has been utilized in boundary layer flows[1,2] and for the hypersonic viscous shock-layer problem.[3]

Two methods have been developed for generating the surface coordinates. The first method uses the orthogonal condition in finite-difference form to determine the surface coordinates with the metric coefficients and curvature of the coordinate lines calculated numerically. The second method obtains analytical expressions for the metric coefficients and for the curvature of the coordinate lines. Both methods assume the body surface is defined in terms of a cylindrical coordinate system where $r = r(x, \phi)$. The surface inclinations δ_1 and δ_2 as illustrated in Figure 2 are determined from

$$\tan \delta_1 = \left(\frac{\partial r}{\partial x}\right)_\phi \quad \text{and} \quad r \tan \delta_2 = -\left(\frac{\partial r}{\partial \phi}\right)_x$$

and are known quantities.

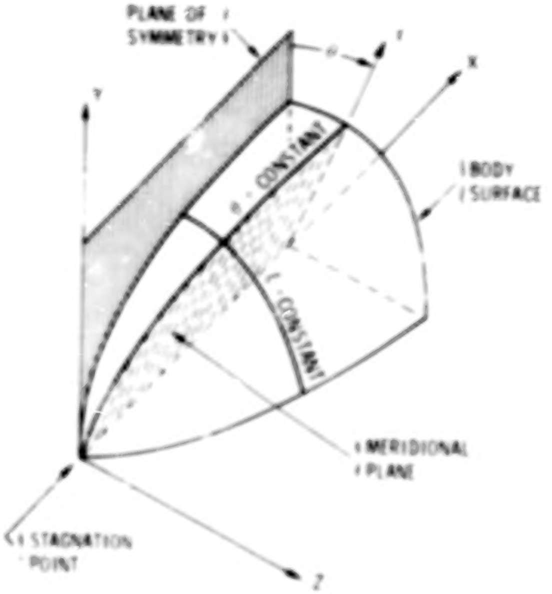


Figure 1. Surface Coordinate System.

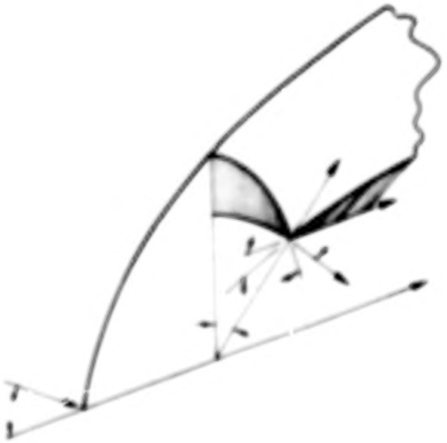


Figure 2. Angles δ_1 and δ_2 Defined in the Cylindrical Coordinate System.

---

* This work was supported by the U. S. Department of Energy under contract DE-AC04-76-DP00789.

† A U. S. Department of Energy Facility.

In the numerical method,[1,2] the orthogonal condition for the surface coordinates results in the relation

$$dx = \lambda \, d\theta \qquad \text{(along } \xi = \text{constant)}$$

where

$$\lambda = r \tan \phi_1 \tan \phi_2/(1 + \tan^2 \phi_1)$$

The equation of the surface provides the relation

$$dr = \tan \phi_1 \, dx - r \tan \phi_2 \, d\theta$$

The surface coordinate ξ is determined numerically from the foregoing equations by assuming a value of $d\theta$ and marching away from $\theta = 0$ to determine the values of x and r. In addition the metric coefficients are determined numerically from

$$h_\xi = ds/d\xi$$

$$h_\omega = dt/d\omega$$

where

$$\omega = \theta/2\pi$$

$$ds^2 = dx^2 + dr^2$$

$$dt^2 = ds^2 + r^2 d\theta^2$$

The curvature of the coordinate lines is determined from

$$K_\xi = \frac{1}{h_\xi h_\omega} \frac{\partial h_\xi}{\partial \omega} \quad \text{for } \omega = \text{constant}$$

$$K_\omega = \frac{1}{h_\xi h_\omega} \frac{\partial h_\omega}{\partial \xi} \quad \text{for } \xi = \text{constant}$$

with the derivatives replaced with midpoint difference relations.

In the second method[1], an analytical expression is developed for h_ω as follows:

$$h_\omega = 2\pi r \, (1 + \cos^2 \phi_1 \tan^2 \phi_2)^{1/2}$$

A differential equation results for the other metric coefficient as follows:

$$\frac{1}{h_\xi} \frac{dh_\xi}{d\omega} = 2\pi r \cos^2 \phi_1 \tan \phi_2 \left(\frac{\partial \phi_1}{\partial x}\right)_\theta$$

This equation is integrated along ξ = constant lines on the surface from the initial condition $h_\xi = 1$ at $\theta = 0$. The substitution of foregoing equations into the equations for K_ξ and K_ω give analytical expressions for the curvature of the coordinate lines. In evaluating these relations, the variations of x and θ along the ξ = constant coordinate must be known.

A sphere at angle of attack as shown in Figure 3 is used to illustrate the computation of the surface coordinates with both methods. The surface coordinates on the sphere as viewed from the side are illustrated in Figure 4. The ξ = constant lines result from planes intersecting the sphere with these planes passing through the line which is normal to the plane of symmetry and is located at

$$x/a = \sqrt{1 - (b/a)^2}$$

$$y/a = (x/a)^2/(b/a)$$

The metric coefficients for this coordinate system are given in Figures 5 and 6 with good agreement between the two methods. The curvature of the coordinate lines is given in Figures 7 and 8. It is noteworthy that K_ξ is independent of ξ. The differences evidenced in Figure 8 can be partially attributed to the numerical evaluation of K_ω being at one-half mesh space locations away from the ξ indicated.

The numerical method of generating the orthogonal surface coordinates has been applied to ellipsoids, paraboloids and elliptic-paraboloids. The coordinates on an ellipsoid are illustrated in Figure 9. The second method or analytical approach has only been developed for the sphere.

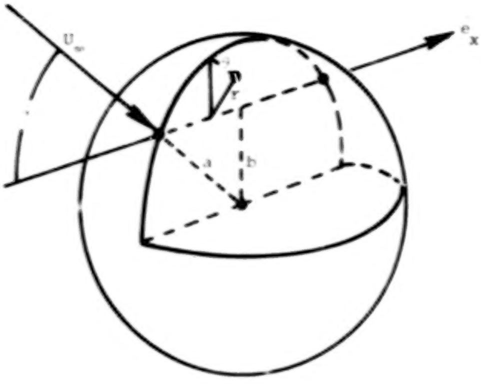


Figure 3. Cylindrical Coordinate System for "Sphere at Angle of Attack".

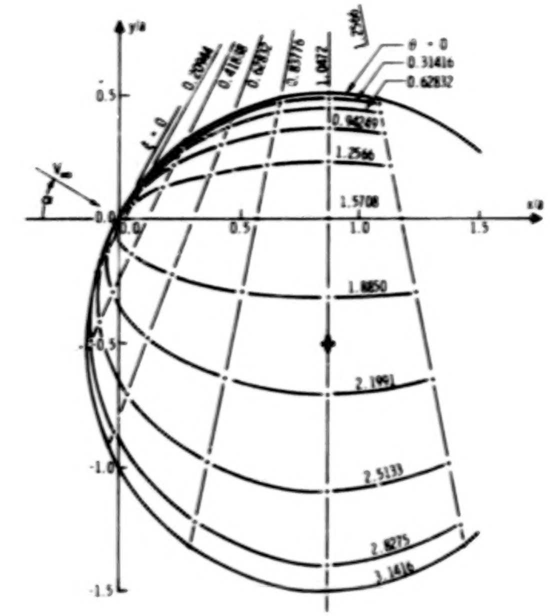

y/a
x/a
0.5
0.0
-0.5
-1.0
-1.5
0.0
0.5
1.0
1.5
ξ = 0
0.20944
0.41888
0.62832
0.83776
1.0472
1.2566
θ = 0
0.31416
0.62832
0.94248
1.2566
1.5708
1.8850
2.1991
2.5133
2.8275
3.1416
V∞
α

Figure 4. Coordinates on a Sphere (α = 30°)

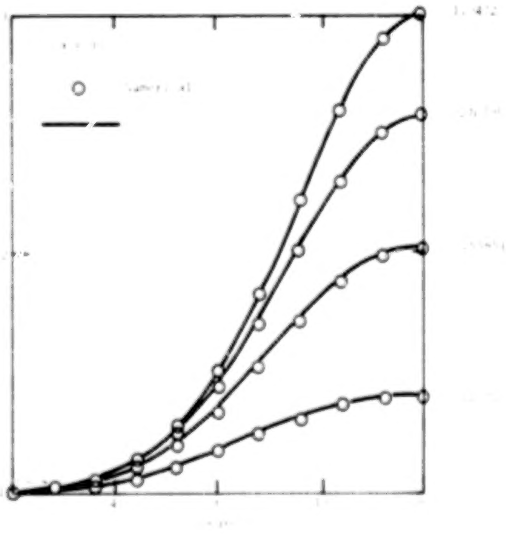


Figure 5. Variation of Metric Coefficient h_ξ.

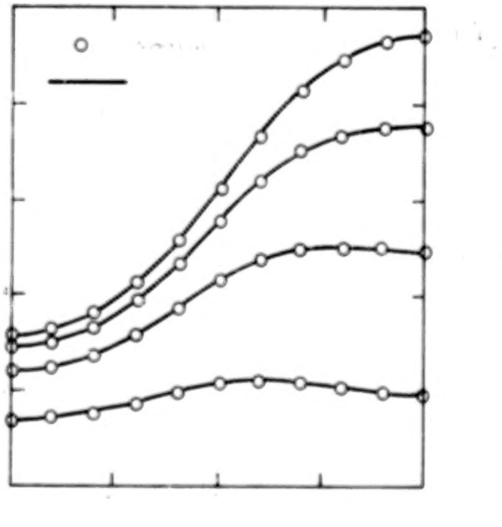


Figure 6. Variation of Metric Coefficient h_ω.

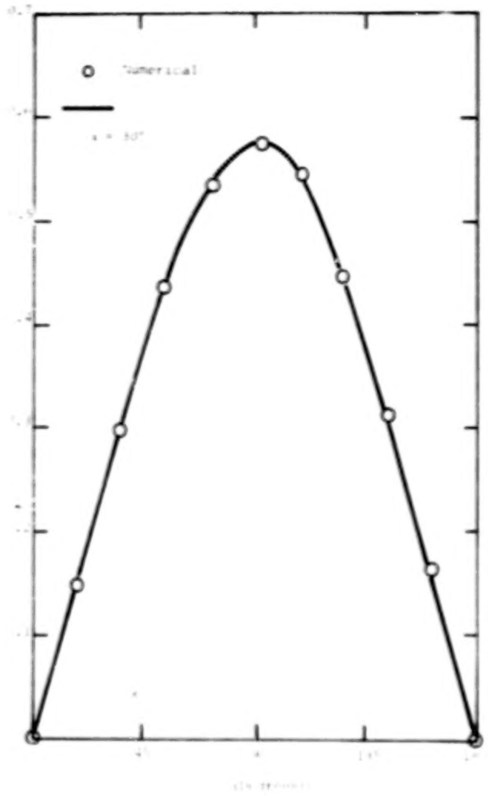


Figure 7. Geodesic Curvature of Lines of Constant θ.

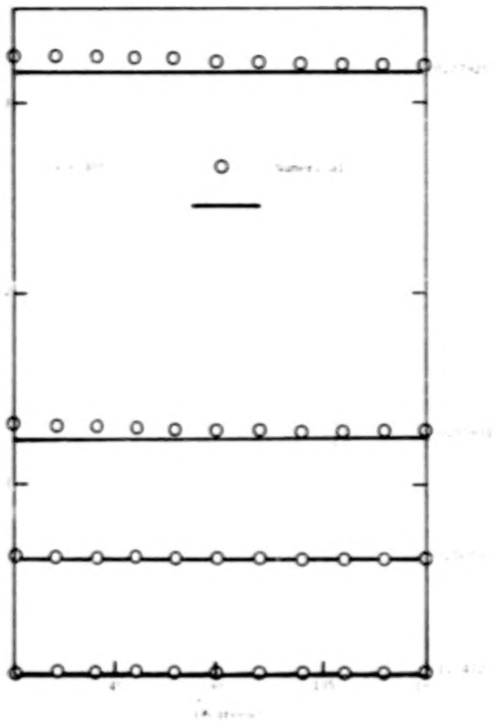


Figure 8. Geodesic Curvature of Lines of Constant ξ.

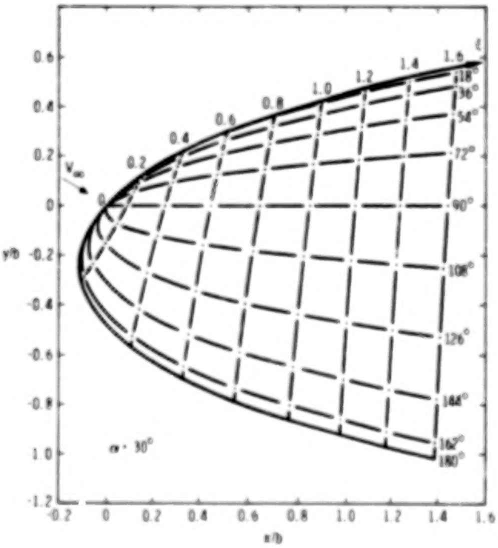

0.6
0.4
0.2
0
y/b
-0.2
-0.4
-0.6
-0.8
1.0
-1.2
-0.2
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
x/b
ξ
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
18°
36°
54°
72°
90°
108°
126°
144°
162°
180°
V_∞
α = 30°

Figure 9. Surface Coordinates on Ellipsoid (b/a = 1/4)

References:

1. F. G. Blottner and Molly A. Ellis, "Finite-Difference Solution of the Incompressible Three-Dimensional Boundary Layer Equations for a Blunt Body," Computers and Fluids, Vol. 1, pp 133-158 (1973).

2. F. G. Blottner and Molly A. Ellis, "Three-Dimensional, Incompressible Boundary Layer on Blunt Bodies. Part I: Analysis and Results," Sandia Laboratories Research Report SLA-73-0366, April 1973.

3. J. B. Moreno, "Formulation of the Three-Dimensional Hypersonic Viscous Shock-Layer Problem Part I: Governing Equations and Geometrical Formulas," Sandia Laboratories Research Report SLA-73-0030.

An Adaptive Computation Mesh for the Solution of Singular Perturbation Problems

J. U. Brackbill and J. Saltzman
Courant Institute of Mathematical Sciences

An adaptive mesh for singular perturbation problems in two and three dimensions is reported. The adaptive mesh is generated by the solution of potential equations which are derived by minimizing the integral I, written,

$$I = \int_D [\{(\nabla_\xi)^2+(\nabla_\eta)^2\}+\lambda_v\{wJ\}+\lambda_o\{(\nabla_\xi\cdot\nabla_\eta)^2\}]dV \qquad (1)$$

where $x(\xi,\eta), y(\xi,\eta)$ represent a mapping from a parameter space P, $0 \leq \xi \leq I$, $0 \leq \eta \leq J$, where $w(x,y) > 0$ is given, λ_v and λ_o are nonnegative constants, and J, the Jacobian, is written,

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} \quad . \qquad (2)$$

In the usual manner, the mesh is constructed by joining points in (x,y) corresponding to integer values of ξ and η by straight lines to form a net of arbitrarily shaped, quadrilateral cells (1).

The variational formulation is equivalent to Winslow's method (2) when λ_o and λ_v are zero. The Euler equations are those given by Winslow, and their solution maximizes the smoothness of the mapping. The additional terms modify other attributes of the mapping in a similar way. When $\lambda_o > 0$, the mapping is modified to make it more orthogonal. When $\lambda_v > 0$, the mapping is modified to make wJ^2 more nearly constant over the mesh. By choosing w(x,y) appropriately, and with λ_v , $\lambda_o > 0$, the zone size variation and skewness can be controlled.

In singular perturbation problems, control of zone size variation can affect the effort required to obtain accurate, numerical solutions of finite difference equations. Consider a simple, difference approximation,

$$\left\langle \frac{df}{dx} \right\rangle = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} \quad , \qquad (3)$$

where i corresponds to ξ , $0 \leq i \leq I$. In the usual manner, the truncation error is written,

$$\varepsilon = \frac{1}{2}\frac{d^2f}{dx^2}x_{\xi\xi} + \frac{1}{6}\frac{d^3f}{dx^3}(x_\xi)^2 + \ldots \quad . \tag{4}$$

When f is sufficiently smooth, ε is least when $x_{\xi\xi} = 0$. However, when f is given, for example,

$$f = (1 + \exp(x/\delta))^{-1} \quad , \tag{5}$$

for which $d^nf/dx^n = O(\delta^{-n})$ is not finite for $\delta = 0$, the error ε is bounded only if $(x_\xi/\delta) < 1$ in the interval $-\delta < x < \delta$. An equally spaced mesh with x_ξ sufficiently small satisfies this requirement, but one with $x_\xi \lessapprox \delta$ only when $-\delta < x < \delta$, and larger everywhere else satisfies it with fewer mesh points. With w(x) given by,

$$w(x) = \left| \frac{1}{f}\frac{df}{dx} \right| \quad , \tag{6}$$

minimizing the integral in Eq. 1 causes the mesh spacing to approach that given by the equation,

$$\left| \frac{1}{f}\frac{df}{dx} \right| x_\xi = \text{const.} \quad , \tag{7}$$

as λ_V increases. As a result, where w is largest the mesh spacing is smallest, and vice versa.

Numerical results for a singular perturbation problem in two dimensions are shown in the accompanying figures. A steady solution to the equation,

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\underset{\sim}{u}\phi) = \kappa\nabla^2\phi \quad , \tag{8}$$

is sought for small values of κ with u given. Such solutions are obtained when the diffusion and convective transport are in balance everywhere,

$$\underset{\sim}{u} = \frac{\kappa}{\phi}\nabla\phi \quad . \tag{9}$$

When $\underset{\sim}{u}$ is given by,

$$\underset{\sim}{u} = -\frac{1}{\kappa}(1+\exp((r-r_o)/\kappa))^{-1}(1+\exp(-(r-r_o)/\kappa))^{-1} \quad , \tag{10}$$

the convective transport term is significantly different from zero in an annulus of width κ with radius r_o. When $w(r,\theta)$ is given by,

$$w(r,\theta) = \underset{\sim}{u}\cdot\underset{\sim}{u} \quad , \tag{11}$$

the zones of the computation mesh are made smaller where $|\underset{\sim}{u}|$ is largest as shown in Fig. 1.

In Fig. 2, the error in the numerical solution of Eq. 8 on the adaptive mesh as measured by the maximum norm,

$$\varepsilon_{max} = \underset{i,j}{\mathrm{Max}} \left| \underset{\sim}{u} - \langle \frac{1}{6} \nabla\phi \rangle \right| \quad , \tag{12}$$

is compared with the error on a fixed rectilinear mesh. The accuracy obtained by adapting the mesh can be obtained, in most cases, only by a threefold refinement of the regular mesh in each coordinate direction.

The adaptive mesh has been used in calculations of resistive magnetohydrodynamic flow in two dimensions with the weight function,

$$w = (\nabla \times \underset{\sim}{B}/\underset{\sim}{B})^2 \quad .$$

The results indicate a significant increase in the maximum, representable magnetic Reynolds number. The adaptive mesh can be applied easily to other fluid flow problems with the appropriate choice of weight function.

The use of the adaptive mesh in time dependent flow problems will be discussed, and results will be presented.

1. Joe F. Thompson, Frank C. Thames and C. Wayne Mastin, J. Comp. Phys. 15, 299 (1974).
2. A. M. Winslow, J. Comp. Phys. 1, 149 (1966).

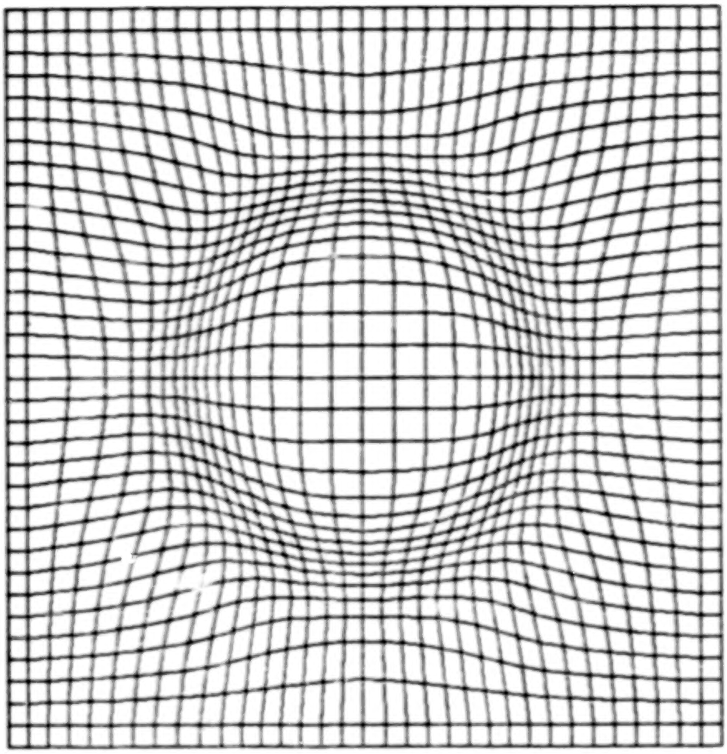


Figure 1.- An adaptive mesh with r_o equal to 1/4 and κ equal to 1/40 the mesh width. The cells are concentrated in the region of maximum gradient.

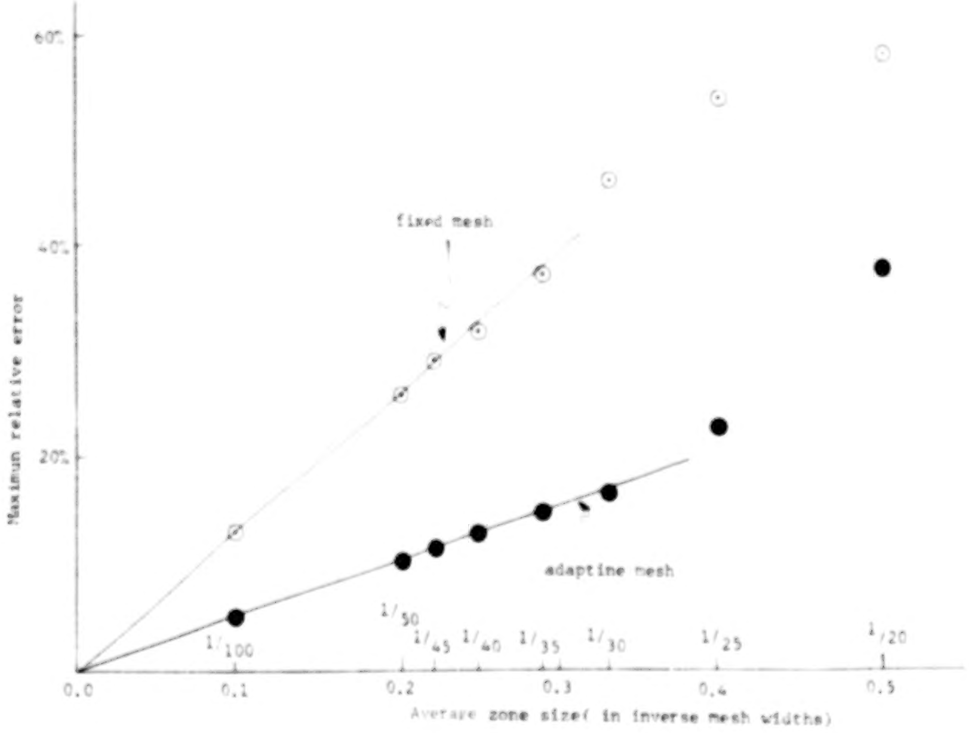


Figure 2.- Similar scaling of the maximum relative error with zone size obtained for both adaptive meshes (like the one shown in Fig. 1) and fixed meshes, but many fewer zones are required with an adaptive mesh.

A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS

J. E. Carter, D. E. Edwards, and M. J. Werle
United Technologies Research Center
East Hartford, Connecticut

ABSTRACT

A new self adaptive coordinate transformation for the finite-difference solution of turbulent boundary-layer flows is presented which permits a uniform mesh to be used in the computational coordinate which extends across the layer. This coordinate transformation uses the local value of the skin friction coefficient to scale the thickness of the wall layer region, and the local maximum value of turbulent viscosity to scale the boundary-layer thickness. Results are presented for two dimensional boundary layers in both positive and negative pressure gradients and comparisons are made with experimental data and conventional variable-grid results for low-speed turbulent boundary-layers. The cases chosen illustrate the capability of this new transformation to capture the boundary layer growth over the full extent of laminar, transitional, and turbulent flow with no grid adjustment as well as its ability to consistently enlarge the wall layer region for accurate shear stress representation. In addition, preliminary results of mesh refinement studies using the new coordinate transformation are presented.

Figure 1. Introduction

Current procedures which are used to generate the mesh across a turbulent boundary layer require the specification of several mesh parameters which are generally difficult to relate to the length scales of the flow itself. In addition, these length scales vary as the solution evolves downstream thereby resulting in a mesh which although "optimum" in one region, may be inappropriate in another. The objective of the present investigation is to develop a procedure which simplifies the specification of the grid point distribution across the turbulent boundary layer. It is desired to have this procedure properly account for the growth of the wall layer as well as the overall boundary-layer thickness. Since most flows are initially laminar at the start of the boundary layer and then are followed by transition to turbulent flows then this procedure should be uniformly applicable to laminar, transitional, and turbulent flows. The approach taken is to develop an adaptive grid technique based on known analytical properties of boundary layer flows. This approach results in a coordinate transformation which is based entirely on fluid dynamic concepts.

Objective

- **Develop a procedure which:**
  - **Simplifies specification of grid point distribution across turbulent boundary layer**
  - **Properly accounts for wall layer and boundary layer thicknesses**

Approach

- **Adaptive grid technique based on known analytical properties of boundary layer flows**

Figure 2. Grid Requirements for Turbulent Boundary Layers

It is well known that turbulent boundary layers are characterized by two transverse length scales, the boundary layer thickness and the wall layer thickness. These two length scales generally are quite different in magnitude thereby making the analysis of turbulent layers more complicated than laminar boundary layers where generally only one length scale is present, the boundary layer thickness. In addition the wall layer and boundary layer thicknesses vary in the stream direction depending upon the pressure gradient, wall boundary conditions, etc. In laminar flow it has been shown that when the boundary layer equations are expressed in terms of the Levy-Lees variables, the streamwise growth of the boundary layer is significantly reduced thereby simplifying the numerical solution of the governing equations. Most turbulent analyses also use the Levy-Lees variables but since these variables do not properly capture the boundary layer thickness it is necessary to monitor the numerical solution and add points in the outer region to accommodate the boundary layer growth. Also, in order to provide adequate resolution of the wall and wake region and simultaneously use as few grid points as possible, practically all numerical approaches for turbulent boundary layers use a fine mesh near the wall and a coarser mesh in the outer region as shown here. There are two difficulties with this approach: 1) the initial choice of the mesh distribution, and 2) the adjustment of this mesh as the wall and boundary layer thicknesses vary downstream. A new coordinate transformation was devised to simultaneously capture the boundary layer growth and automatically scale the inner wall layer region thereby allowing a uniform step to be used in the transformed coordinate, N. The resulting turbulent profile, schematically shown here, has the appearance of a laminar profile when plotted in terms of N.

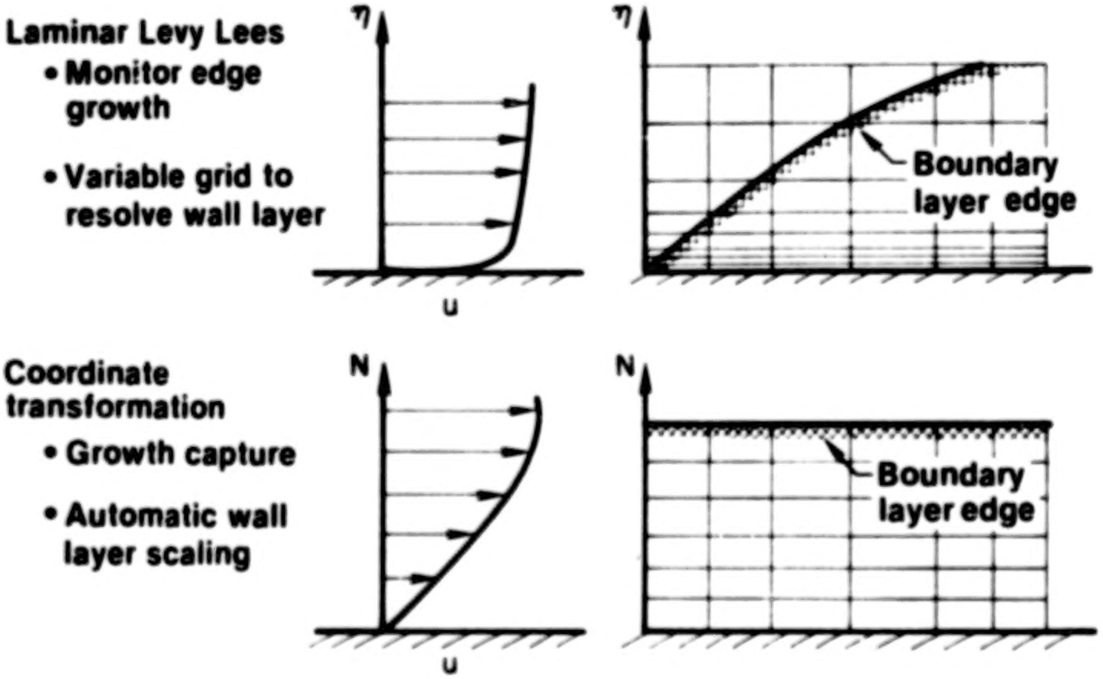


Figure 3. Capture of Turbulent Boundary Layer Growth

The development of the coordinate transformation begins by first generalizing the Levy-Lees transformation for laminar flow to turbulent flow by using a reference turbulent viscosity level to replace the laminar edge viscosity in these transformed variables. For laminar flows the usual Levy-Lees transformation converts the equations from physical variables to similarity variables such that even when the flow is not self-similar the boundary layer edge is essentially constant in the transformed normal coordinate. In the laminar case the normalized molecular viscosity coefficient is O(1) in the outer region of the boundary layer. For the turbulent case this transformation is modified to normalize the turbulent viscosity coefficient to O(1) in the outer region but is done in such a manner that the form of the equation is unchanged from the laminar set. Thus in these transformed variables, in the outer region, the solution for laminar and turbulent flow should be approximately the same since the outer boundary condition (F=1) is the same for both. Therefore, since these variables capture the boundary layer growth in laminar flow, the same growth capture should occur in the turbulent case. The turbulent Levy Lees transformation is a generalization of that used by Schlichting in his Ph.D. thesis in 1930 to transform the turbulent momentum equation for jets and wakes into a "laminar-like" form thereby permitting the laminar similarity solution to be used for a turbulent flow. The new turbulent Levy Lees transformation can be used with any turbulence model provided that a representative turbulent viscosity level can be identified. In the present work the two-layer algebraic eddy viscosity model of Cebeci and Smith was used, in which the reference turbulent viscosity is that for the outer layer. This value varies only with the distance along the surface since an intermittency function was not used at the boundary layer edge.

$$\xi = \int_0^s \rho_e \mu_{t_e} u_e \, ds \qquad \eta = \frac{\rho_e u_e}{\sqrt{2\xi}} \int_0^n \frac{\rho}{\rho_e} \, dn$$

$$\mu_{t_e} = \mu_e \left(1 + \frac{\epsilon}{\mu}\right)_{ref}$$

$\epsilon_{ref} = 0$ **Laminar Levy Lees transformation**

$\epsilon_{ref} \neq 0$ **New turbulent Levy Lees transformation**

$$F = \frac{u}{u_e} \qquad V = \frac{2\xi}{\rho_e \mu_{t_e} u_e} \left(\frac{\rho v}{\sqrt{2\xi}} + F \eta_s \right) \qquad \beta = \frac{2\xi}{u_e} \frac{du_e}{d\xi}$$

Continuity: $2\xi F_\xi + F + V_\eta = 0$

Momentum: $2\xi F F_\xi + V F_\eta = \beta \left(1 - F^2\right) + \left[\frac{\rho \mu_t}{\rho_e \mu_{t_e}} F_\eta\right]_\eta$

where $\mu_t = \mu\left(1 + \frac{\epsilon}{\mu}\right)$

Figure 4. Composite Coordinate Transformation

The use of the turbulent Levy Lees transformation avoids the need to continuously add or subtract grid points at the edge of the turbulent boundary layer due to boundary layer growth or decay. However, a variable grid distribution is still required, in fact now even more so, to adequately resolve the wall layer thickness since it has been correspondingly reduced along with the boundary-layer thickness. Clearly, an inner region transformation is needed to enlarge, in the computational coordinate, the high gradient wall region. Fortunately, the analytical behavior of the turbulent boundary layer profile is known in the wall layer region and this information can be used as the basis for an inner region (wall layer) transformation. It has been established numerous times both analytically and experimentally over the past 40 years that the velocity varies with the logarithm of the distance normal to the wall in the wall region. This relationship is not valid in the immediate vicinity of the wall since it is singular and must be replaced with the laminar sublayer profile where the velocity varies linearly with the distance normal to the wall. Hence a logarithmic coordinate transformation could not be used if we want to solve the equations all the way to the wall, which is desired in most boundary-layer analyses. In a recent paper, Whitfield presented a new analytical expression for the velocity profile in the wall region which also has the proper analytical behavior in the laminar sublayer. This analytical expression is used in the present work in the wall or inner region such that a constant increment in the transformed coordinate results in a constant increment in velocity. With the inner region coordinate transformation established it is now necessary to specify a suitable transformation for the outer or wake region. The outer coordinate transformation is motivated by the observation that with the turbulent Levy Lees transformation discussed in figure 3, the boundary layer edge is fixed and the governing equations closely resemble the laminar equations in the outer region. Hence the outer transformation is deduced from a function which closely fits the Blasius solution. Whitfield found that this function closely fits turbulent data in the outer region which supports the idea that in this region the laminar and turbulent solutions resemble each other. A composite transformation is established by combining the inner and outer transformations employing concepts from the method of matched asymptotic expansions. The final result is that the semi-infinite physical space $0 \le y \le \infty$ is mapped into a unit interval $0 \le N \le 1$ in the computational coordinate N, and that the transformation used is based completely on fluid dynamic concepts to assure a universal applicability of the method. A sketch of the inner, outer, and composite functions is shown to illustrate their relative magnitudes.

Inner region:

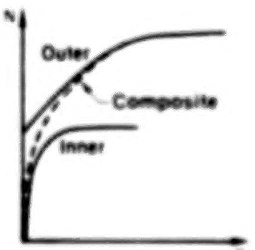
$N_i \sim \frac{u}{u_e}\Big|_i = k_1\sqrt{C_{f_e}}\,\tan^{-1}\left(k_2\sqrt{\xi C_{f_e}}\,\hat{\eta}\right)$

- **Analytical solution by Whitfield**
- **Correct in sublayer,** $u^+ = y^+$

Outer region:

$N_o \sim \frac{u}{u_e}\Big|_o = \tanh^{1/2} a\left(\hat{\eta}+\hat{\eta}_o\right)^2$

- **Normalized viscosity implies outer region is "laminar-like"**
- **Good fit to Blasius solution**
- **Found by Whitfield to match experimental data**



Composite:

$N_c = N_i + N_o - N_{i/o}$

- **Matching condition** $N_{i/o} = N_{o/i}$
- **Solve for** $\hat{\eta}_o$

Figure 5. **Implement Coordinate Transformation**

The composite coordinate transformation presented in figure 4 is incorporated into the boundary-layer equations expressed in terms of the turbulent Levy Lees variables which were discussed in figure 3. The transformed equations are obtained in a straightforward manner and are not substantially different from those in figure 3 other than the explicit dependence of the term $\partial N/\partial \xi$. These equations are solved with a standard implicit finite-difference scheme in which a uniform mesh is used in the normal direction. The use of the coordinate transformation results in less than a 10% increase in computer time over that used by our UTRC computer code which was recently developed using the variable grid finite difference scheme developed by Blottner. This code has been used in the present work to provide calculations for comparison purposes. This new coordinate transformation is an adaptive grid procedure since it relies on two quantities, the local skin friction and the local reference viscosity to complete the specification of the composite coordinate at each streamwise location. In the results presented here these quantities were obtained from the solution at the previous station since a non-iterative scheme was used. This adaptive grid procedure is applicable to laminar flows since the wall layer region is nonexistent (hence $N_i = 0$) and only the outer transformation is used. In transitional flows the wall layer region is initiated at the start of transition and thus allows for the natural development of the wall region as the flow evolves from a laminar to a turbulent boundary layer.

- **Finite difference solution of equations in ξ, N coordinates**
- **Adaptive grid — C_{f_e} and $\left(1+\frac{\epsilon}{\mu}\right)_{ref}$ depend on local solution**
- **Applies to laminar, transitional, and turbulent flow**

  Laminar — set $N_i = 0$

  Transitional — inclusion of inner region initiated at start of transition

  Turbulent — composite transformation

Figure 6. Skin Friction Distribution - Flat Plate

This figure shows a comparison of the skin friction distribution obtained from calculations in which the composite coordinate transformation (adaptive grid) and the variable grid (geometric progression) techniques were used. Both predictions agree well with the experimental data of Wieghardt. In the present case 101 points were used across the layer and there is no plottable difference in the results. Reduction in the number of points from 101 to 21 resulted in essentially the same solution using the adaptive grid; the same reduction for the variable grid scheme resulted in a slightly different solution as shown here. The arrows indicate the locations at which profiles from the different approaches will be compared.

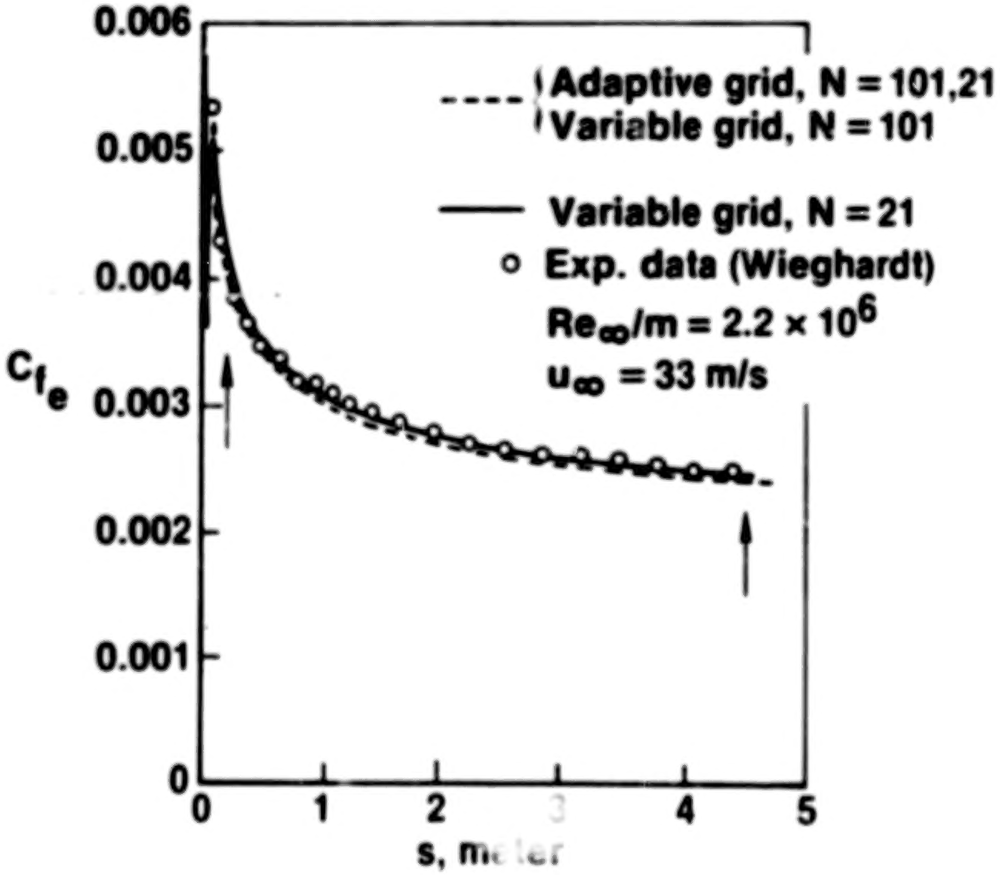


Figure 7. Displacement Thickness Distribution - Flat Plate

Shown here is a comparison of the displacement thickness distributions from the adaptive grid scheme versus that measured experimentally. The agreement is good and the solution is shown to change only a few percent when the grid is reduced. Similar changes were found to occur in the variable grid scheme when the same grid reduction was made. Detailed grid studies are presently underway in order to compare the relative truncation errors of the adaptive grid scheme and the variable grid scheme.

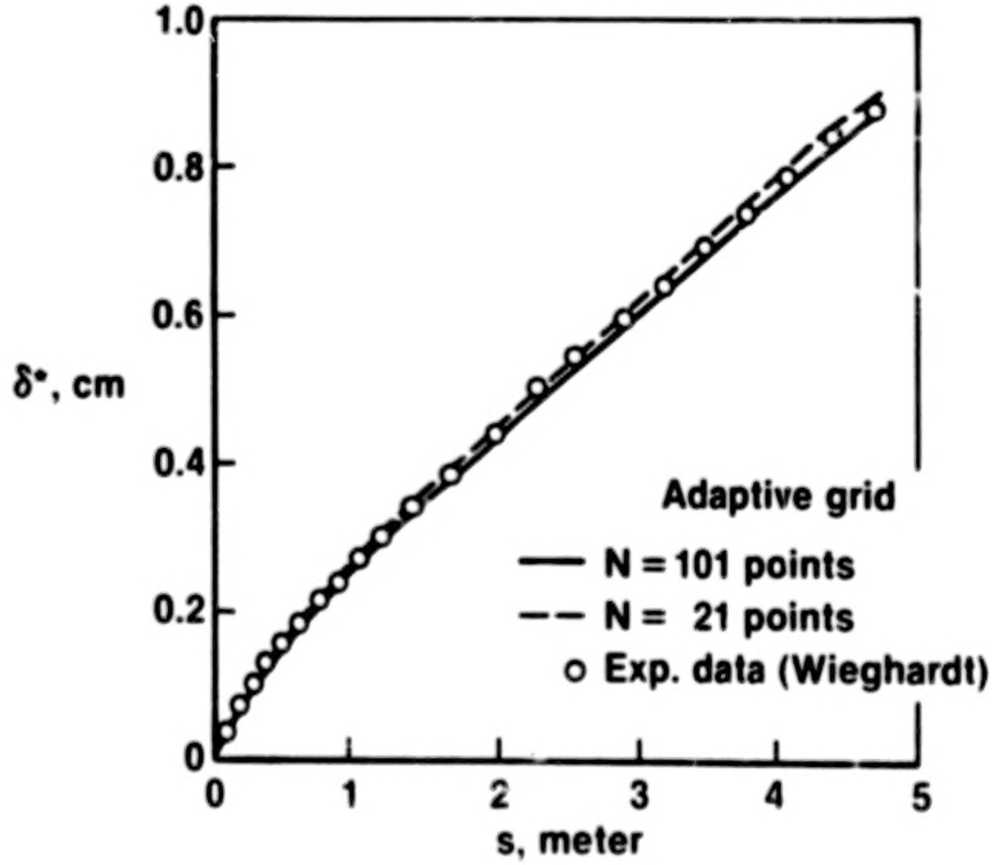


Figure 8. Velocity Profiles in Laminar Levy Lees Variables

In the next several figures flat plate velocity profiles at the locations previously indicated in figure 6 will be shown in terms of the normal coordinate as given by the laminar Levy Lees transformation, the turbulent Levy Lees transformation, and the composite coordinate transformation. The present figure clearly shows the two-layer structure of the turbulent boundary layer as well as the significant boundary layer growth which occurs in this variable. Use of the laminar Levy Lees transformation for turbulent flows is not significantly different than working in the physical or untransformed coordinate. Also plotted is the Blasius solution which is the laminar self-similar solution for a flat plate. Note that the Blasius solution has a much smaller value of η_e at the edge of the boundary layer than the turbulent profiles despite the higher skin friction (slope at wall) in the turbulent case.

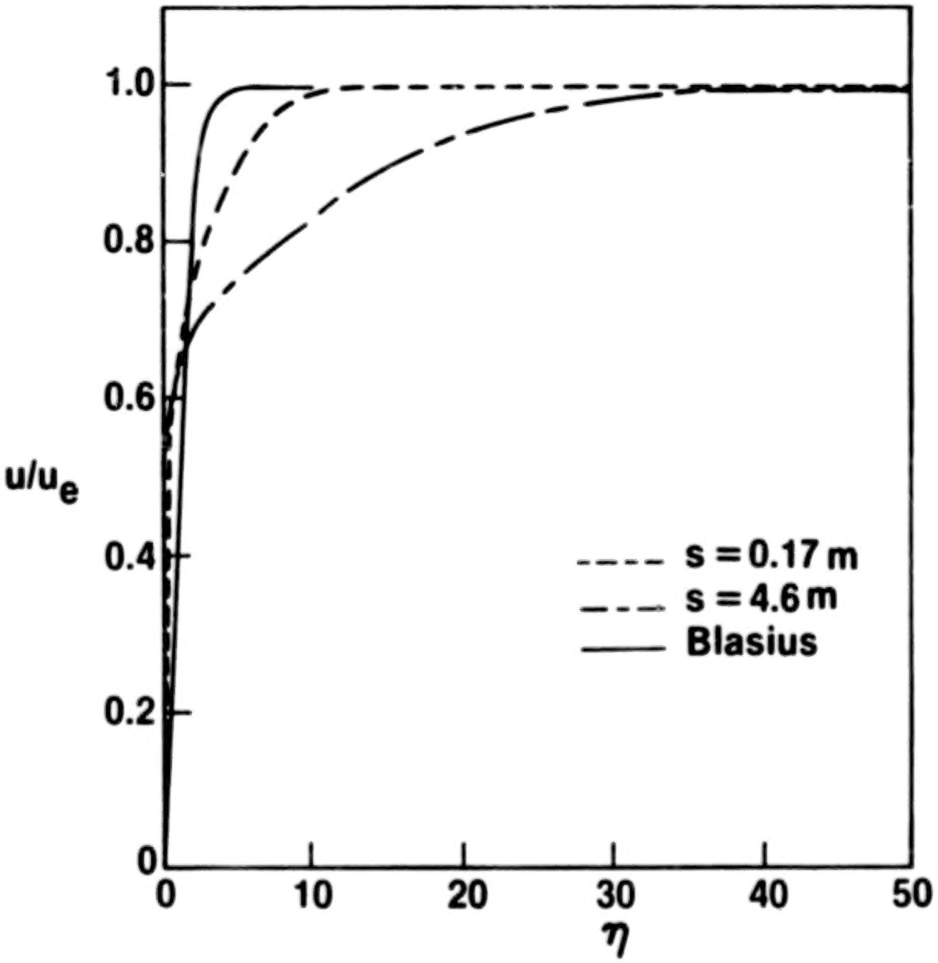


Figure 9. Velocity Profiles in Turbulent Levy Lees Variable

This figure shows the same profiles in the new turbulent Levy Lees variable and indicates that the turbulent boundary-layer thickness has been preserved in this new variable and that it is nearly the same value as that of the Blasius profile. The bar over the η-coordinate is used to distinguish between the turbulent Levy Lees variable and the laminar Levy Lees variable as discussed in figure 3. Both variables have the same form; it is only the interpretation of the ξ-variable contained in the η-variable which distinguishes the two transformations. Despite the capture of the turbulent boundary layer growth, it is seen in this figure that the high gradient wall region still persists which requires a variable grid for adequate resolution.

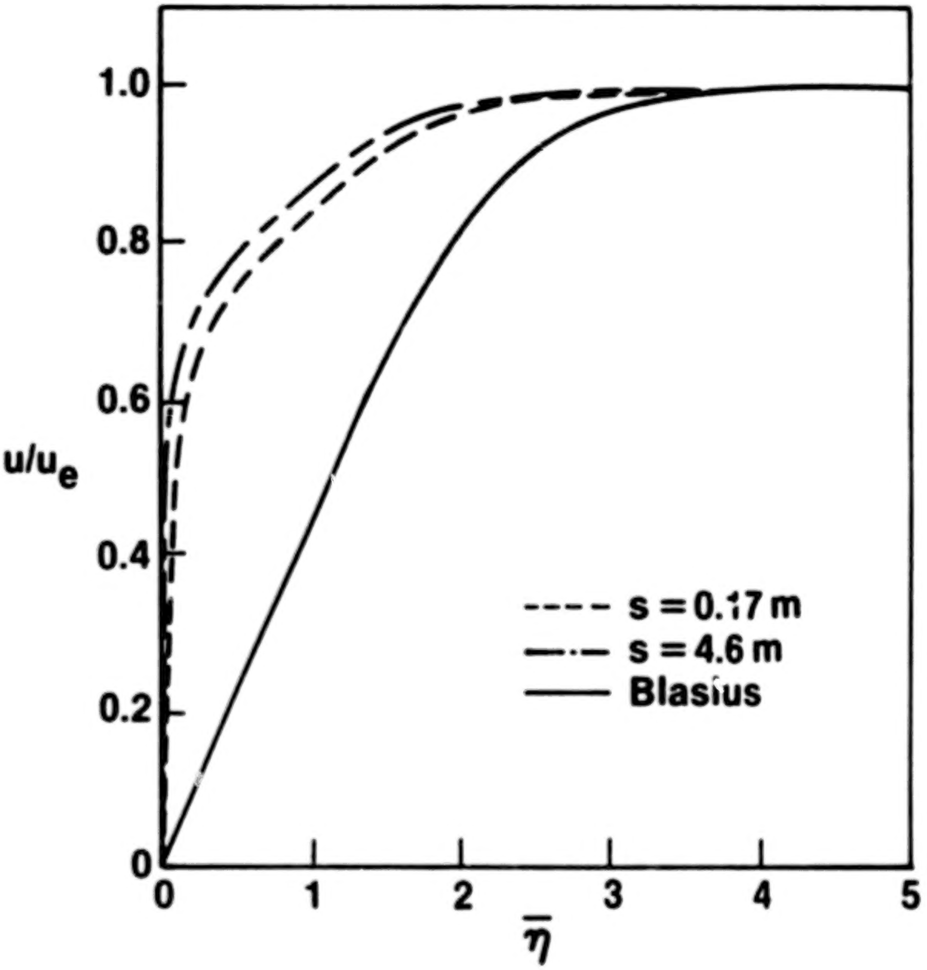


Figure 10. Growth of Boundary Layer Thickness - Flat Plate

This figure shows a comparison between the streamwise variation of the boundary layer edge as deduced in the laminar Levy Lees variable versus that obtained in terms of the turbulent Levy Lees variable. The ability of the turbulent Levy Lees variable to capture the turbulent boundary layer growth is clearly seen here.

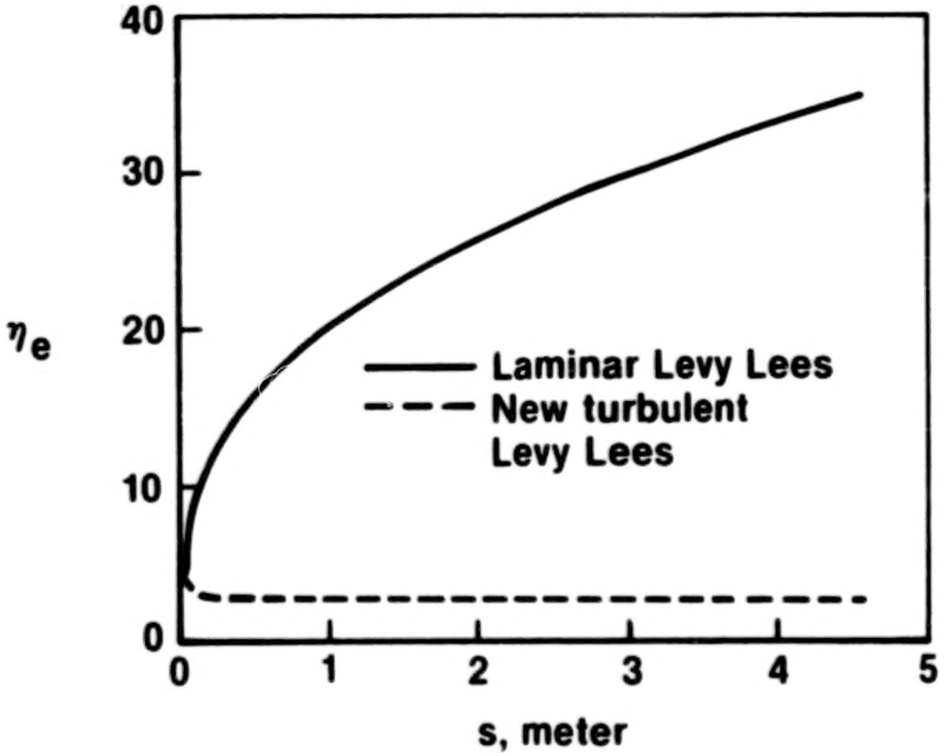


Figure 11. Velocity Profiles in New Composite Coordinate

This figure presents the same profiles shown previously now plotted in terms of the new composite coordinate. It is seen that this transformation results in an enlargement of the wall region, and since the boundary layer edge is captured by the turbulent Levy Lees transformation, the computed turbulent profiles show the same O(1) variation across the layer as the laminar profile thereby permitting a uniform mesh to be used. It is seen that in terms of this new composite coordinate the turbulent profiles change only slightly over a flat plate distance of 4.5M. These changes are greater in the outer region than they are in the inner which is probably due to the more approximate outer coordinate transformation as compared to the use of Whitfield's analytical solution for the inner transformation.

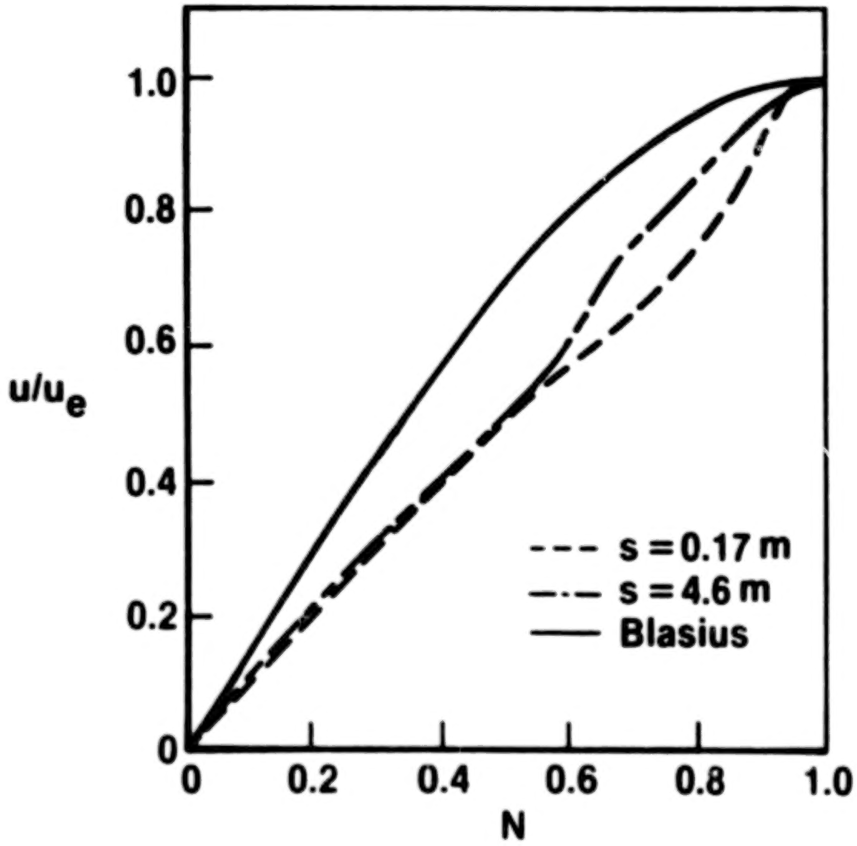


Figure 12. Edge Velocity Distribution

The previous example was a flat plate in which the imposed streamwise pressure gradient is zero. It is well known that boundary layer flows are strongly influenced by the pressure gradient and thus as a test of the new technique presented herein the edge velocity shown in this figure was imposed as a streamwise boundary condition on the boundary layer equations. This distribution was measured by Schubauer and Klebanoff for the airfoil shown here and provides a good test case for the present work since both regions of favorable and adverse pressure gradient are present.

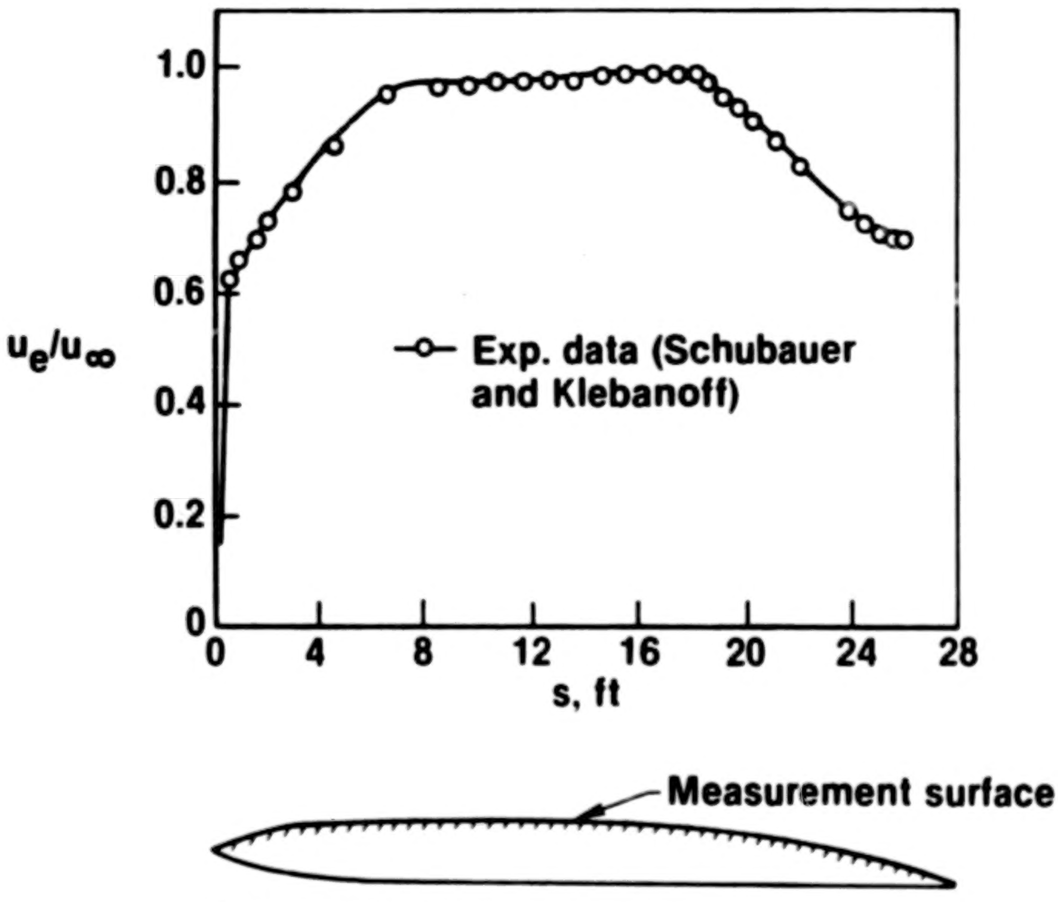
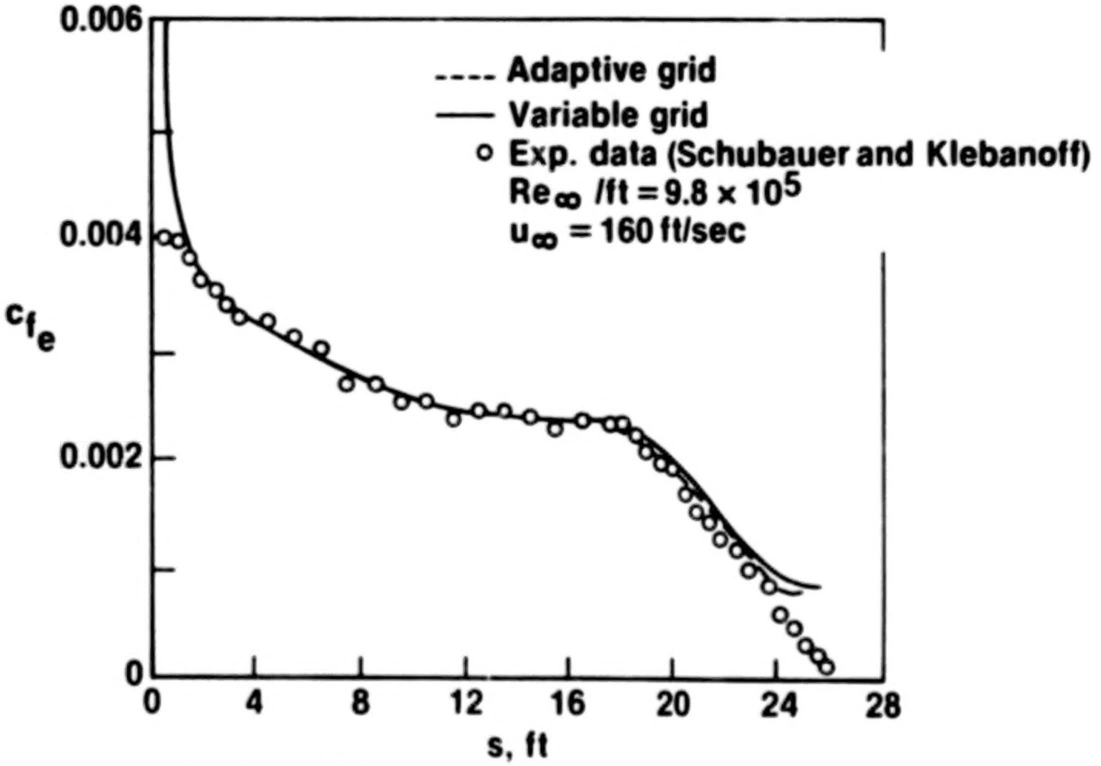


Figure 13. Skin Friction Distribution - Airfoil

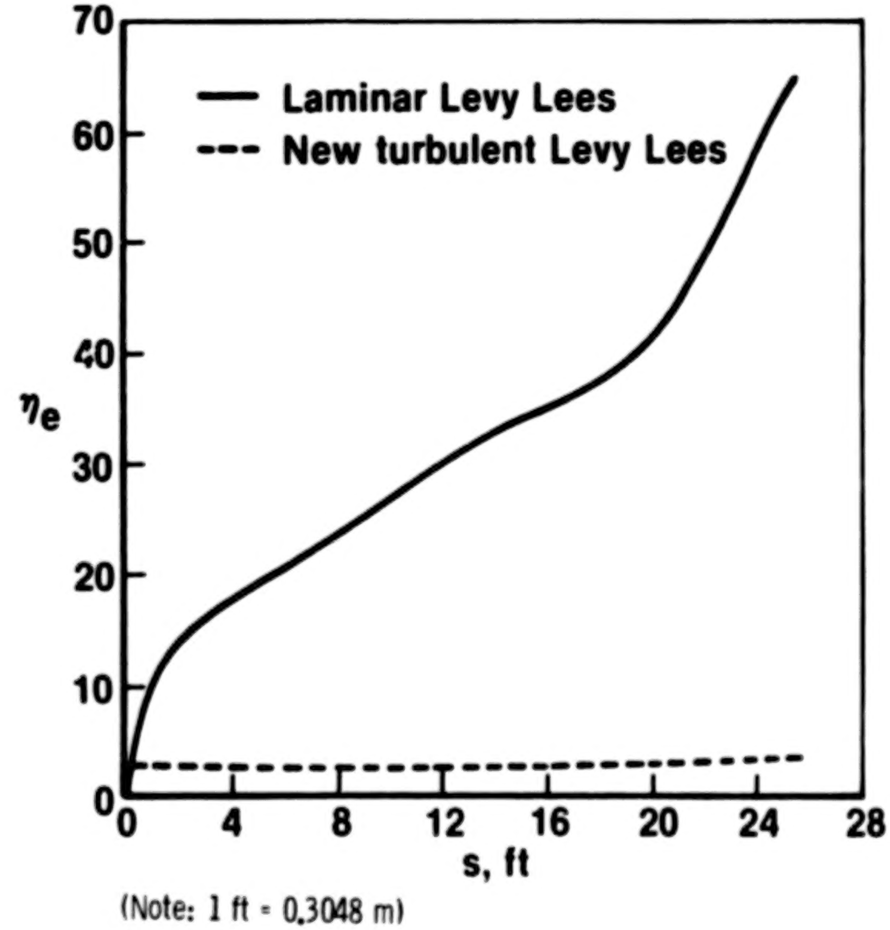
Comparison of the computed skin friction with that measured by Schubauer and Klebanoff is shown here. Excellent agreement is obtained except in the aft strong adverse pressure gradient region where other investigators have concluded that there are three dimensional effects which of course is outside the scope of the present analysis. Comparison of the adaptive grid results with those obtained with the variable grid show that both solutions are the same except in the adverse pressure gradient region where the adaptive grid scheme shows better agreement with the data. Both cases were computed with 101 points across the layer. In the present case the computation does not extend to the separation point so as a further test of the new scheme an analytically imposed edge velocity was prescribed such that separation was encountered. No difficulties were encountered in this case and both the adaptive grid and variable grid schemes yielded nearly the same result.



(Note: 1 ft = 0.3048 m)

Figure 14. Growth of Boundary Layer Thickness - Airfoil

This figure shows that the boundary layer edge is captured with the new turbulent Levy Lees transformation for both positive and negative pressure gradients as was shown in figure 10 for zero pressure gradient. A slight increase in the boundary-layer edge is observed with the turbulent Levy Lees transformation in the adverse pressure gradient region; however, this growth is negligible compared to that which occurs in the usual laminar Levy Lees variable.



(Note: 1 ft = 0.3048 m)

Figure 15. Conclusions

In conclusion, a new adaptive grid procedure has been presented which automatically captures the boundary layer thickness and simultaneously enlarges the wall layer region through the use of a composite coordinate transformation. This new procedure demonstrates the benefit of using fluid dynamic concepts in mesh generation for numerical solutions since scaling problems and singular regions are properly accounted for. The adaptive grid scheme presented here is simpler to use than a variable grid scheme since now only the total number of desired points needs to be specified by the user. In addition, this adaptive grid procedure has been demonstrated to be applicable to laminar, transitional, and turbulent flows.

- **Adaptive grid procedure automatically captures boundary- and wall-layer thicknesses**
- **New procedure demonstrates benefit of incorporating known analytical properties of the flow into mesh generation**
- **Adaptive grid procedure easier to use than variable grid scheme since only total number of points must be specified**
- **Adaptive grid procedure applies uniformly to laminar, transitional, and turbulent flows**

Generation of Orthogonal Boundary-Fitted Coordinate Systems

Roderick M. Coleman

Computation, Mathematics, and Logistics Department
David W. Taylor Naval Ship Research and Development Center
Bethesda, Maryland 20084

ABSTRACT

A method is presented for computing orthogonal boundary-fitted coordinate systems for geometries with coordinate distributions specified on all boundaries. The system which has found most extensive use in generating boundary-fitted grids is made up of the Poisson equations

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P \\ \eta_{xx} + \eta_{yy} &= Q \end{aligned} \qquad (1)$$

The functions P and Q provide a means for controlling the spacing and density of grid lines in the coordinate system. Since all calculations are done in the computational plane, the dependent and independent variables in Equation (1) are interchanged, giving the usual transformed equations

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) &= 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) &= 0 \end{aligned} \qquad (2)$$

where

$$\alpha = x_\eta^2 + y_\eta^2 \qquad \beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^2 + y_\xi^2 \qquad J = x_\xi y_\eta - x_\eta y_\xi$$

The condition for orthogonality, i.e., ξ = constant lines perpendicular to η = constant lines, is $\beta = 0$, because

$$\beta = 0 \rightarrow x_\xi / y_\xi = -y_\eta / x_\eta$$

which is equivalent to

$$1/y_x \Big|_{\eta = \text{constant}} = -y_x \Big|_{\xi = \text{constant}}$$

As a generating system based entirely on β, we consider

$$\beta_\xi = \beta_\eta = 0 \qquad (3)$$

which can have an orthogonal solution only when $\beta = 0$ at the corners of the computational region. An iterative solution of the generating system given in Equation (3) is applied successfully to several geometries. While questions remain concerning the existence and uniqueness of orthogonal systems, the generating method presented here adds to the available, useful techniques for constructing these systems.

Figure 1 provides a comparison of two grids generated for a square region with nonuniform boundary coordinate spacing in both vertical and horizontal directions. The nonorthogonal mesh shown in Fig. 1a was generated using the Poisson system given by Equation (2) with $P \equiv Q \equiv 0$. Equation (2) was replaced with central difference formulae and the resulting system was solved by successive overrelaxation (SOR). The orthogonal mesh shown in Fig. 1b was obtained using Equation (3) as a generating system. Equation (3) was expanded and each derivative was replaced with the appropriate central difference formula. Again, the resulting system was solved by SOR iteration.

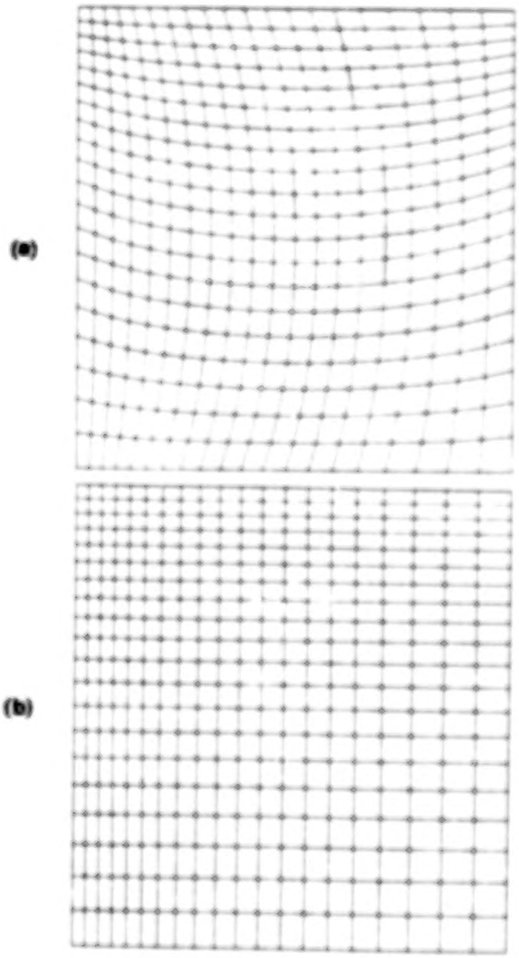


Figure 1

Two 21 x 21 girds generated for a simply-connected region with one convex boundary are shown in Figure 2. Fig. 2a shows a nonorthogonal coordinate system generated by Equation (2) with $P \equiv Q \equiv 0$ (a Laplace system); Fig. 2b shows a coordinate system generated by Equation (3). Note the orthogonality of the coordinate lines intersecting the curved upper boundary in Fig. 2 and the resultant bending of these lines in the interior.

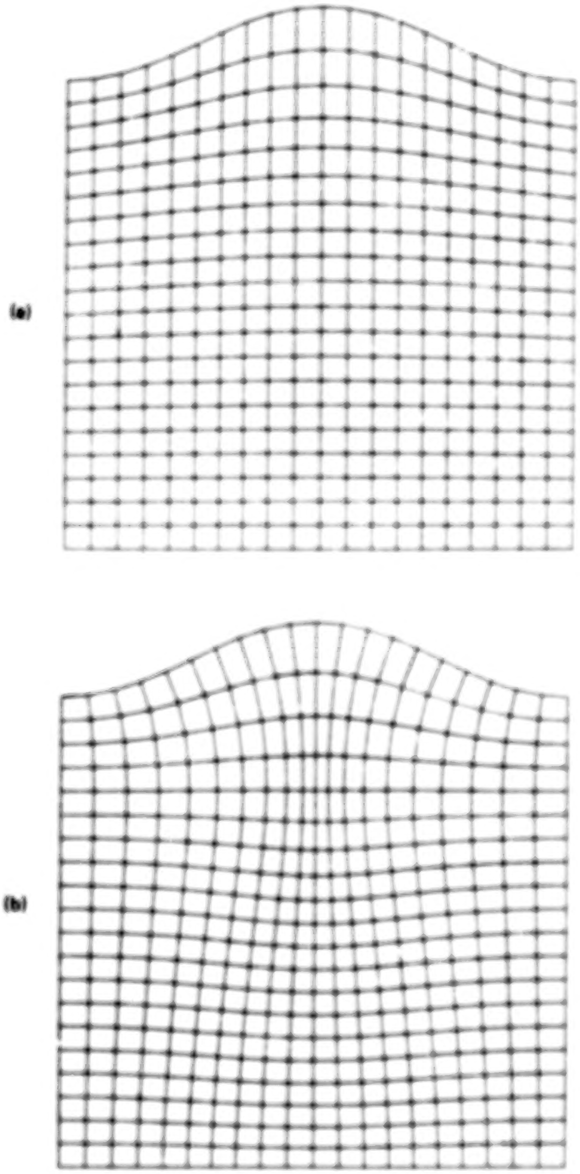


Figure 2

Figure 3 shows a region similar to that of Fig. 2 with a concave rather than convex curved boundary. As before, Fig. 3a shows a Laplace-generated grid and Fig. 3b shows a β-generated grid obtained using Equation (3). The orthogonal mesh must have rather fine spacing near the concave upper boundary to accommodate the curvature. To verify that the fine mesh spacing in Fig. 3b is due to the geometry and not to a singularity in the transformation, we have refined the mesh as seen in the next figure.

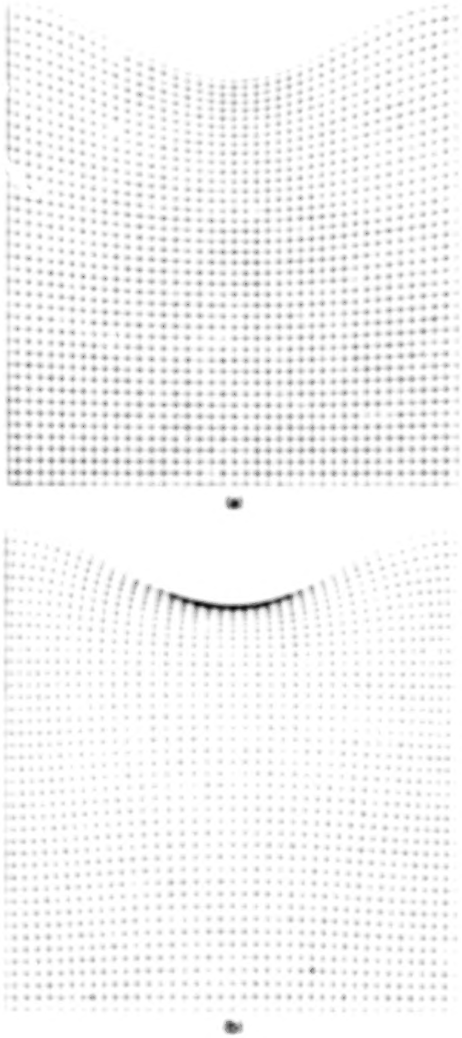


Figure 3

Figure 4 compares two different grids, one coarse with 1681 points and the other fine with 6561 points, generated for the concave region. The fact that corresponding grid lines are in about the same position in both meshes confirms that the coarse discretization yields a good approximate solution to the exact problem. A further confirmation comes from consideration of the Jacobian at the midpoint of the upper boundary. The value of the Jacobian computed on the coarse mesh is nonzero and agrees very well with the value computed on the fine mesh. There is no indication of a zero Jacobian anywhere in the region.

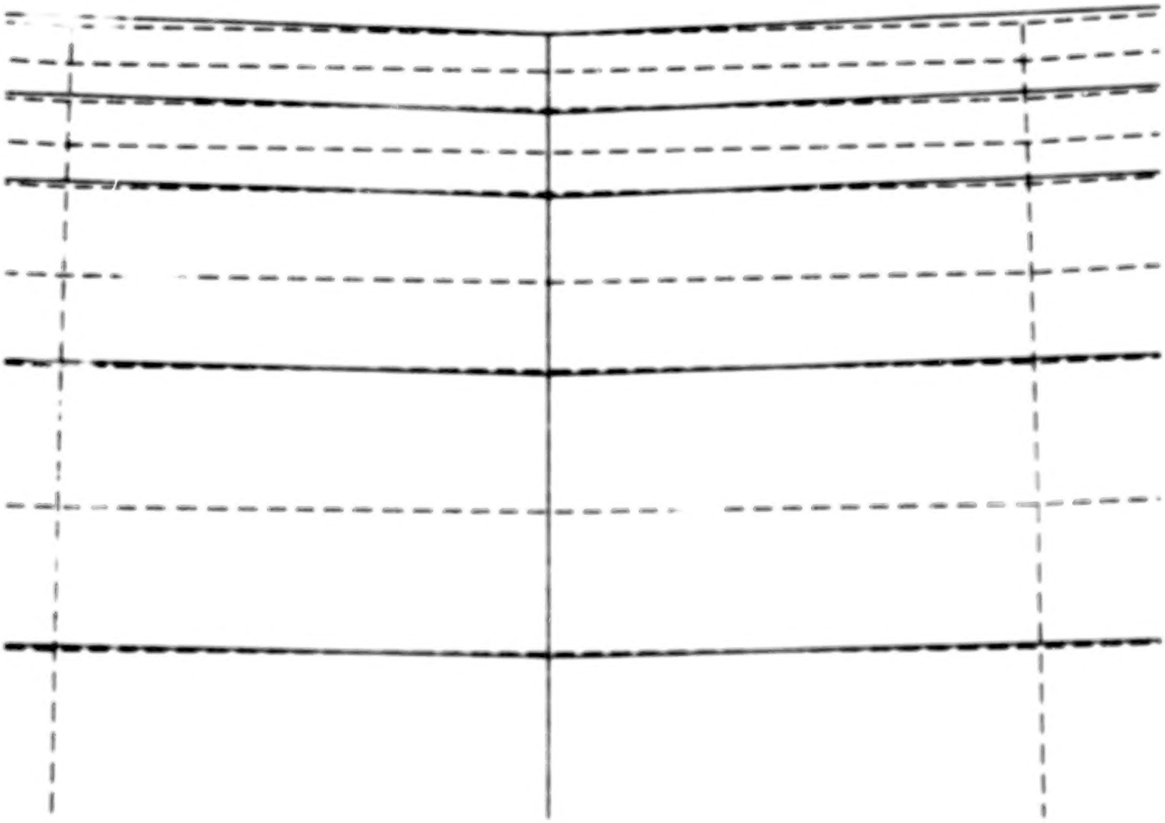


Figure 4

To demonstrate some of the problems that can arise, we attempted to generate an orthogonal mesh on a region similar to the previous one but with greater curvature of the concave boundary. The grid shown in Fig. 5a was generated by a Laplace system and the unacceptable grid in Fig. 5b was generated by the system of Equation (3). To verify that a mesh with crossing lines can also be produced by a Poisson system, we computed directly the forcing functions P and Q using Equation (2) with x and y as given in Fig. 5b. We then solved Equation (2) iteratively for x and y using this P and Q, and regenerated the grid of Fig. 5b.

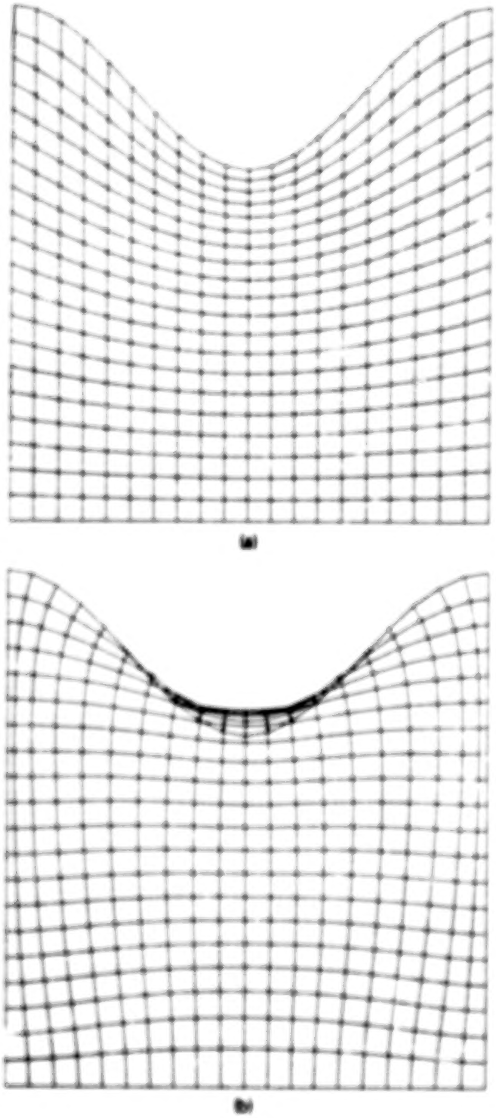


Figure 5

As the final example, we considered a doubly-connected region bounded by concentric circles as shown in Fig. 6. Since this region is symmetric with respect to any line passing through the center, each grid was generated for half the region and reflected in the line of symmetry. The symmetry line was treated as a boundary with fixed coordinate distribution, thus assuring that $\beta = 0$ at the corners of the computational region. The spacing on the outer boundary, but not on the inner boundary, was uniform. Had the spacing on both boundaries been uniform, the grid produced by the Laplace generating system (Fig. 6a) would have been the usual polar coordinate system which is orthogonal. In Figs. 6a and 6b, the line of symmetry was taken as a horizontal line through the center of the figure. The mesh of 6a was used as an initial guess for the iterative procedure used to obtain the mesh of 6b.

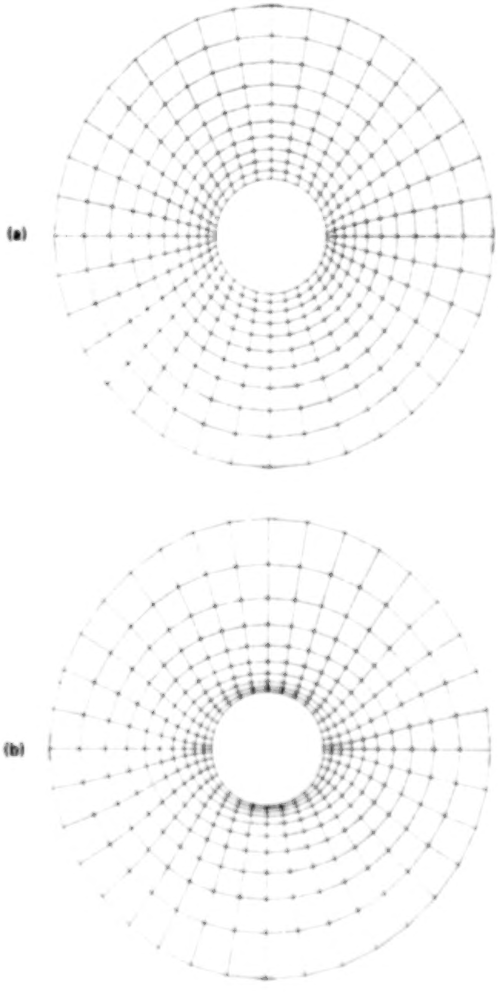


Figure 6

In Fig. 7, we show a β-generated grid computed for the same doubly-connected region used in the previous figure. As before, the mesh of Fig. 6a was used for the initial guess, but in this case the line of symmetry was taken as a vertical line through the center. Interestingly, the two orthogonal grids generated for the same physical region (Figs. 6b and 7) are quite dissimilar because different points were held constant after the same initial guess.

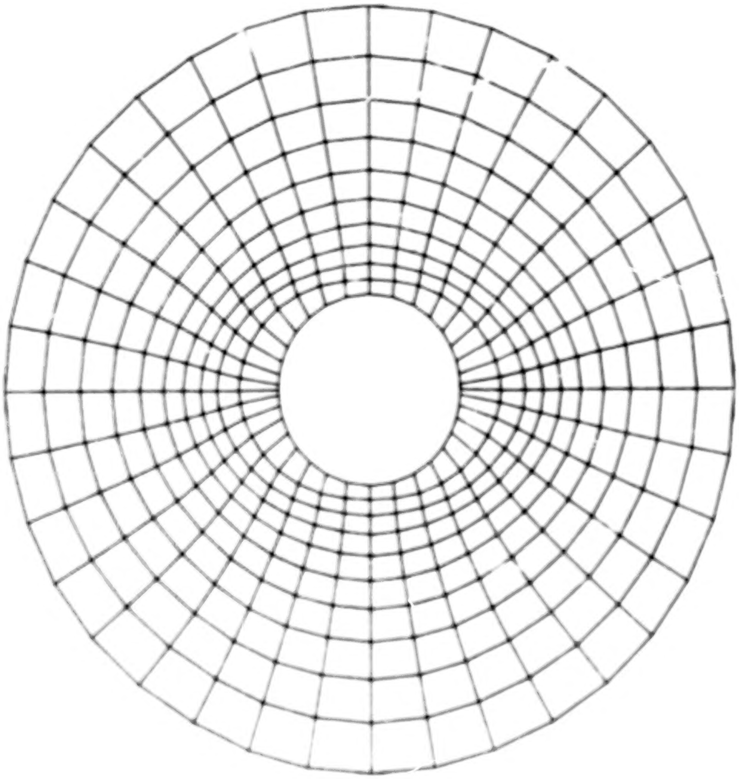


Figure 7

NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS*

S. K. Dey
Department of Mathematics
Eastern Illinois University
Charleston, Illinois 61920

Abstract

Finite difference solution of nonlinear partial differential equations requires discretizations and consequently grid errors are generated. These errors strongly affect stability and convergence properties of difference models. Previously such errors were analyzed by linearizing the difference equations for solutions. In this article properties of mappings of decadence [1,2] were used to analyze nonlinear instabilities. Such an analysis is directly affected by initial/boundary conditions. An algorithm has been developed, applied to nonlinear Burgers' equation [3,4] and verified computationally. A preliminary test shows that Navier-Stokes' equation may be treated similarly.

---

*This work has been supported by Minna-James-Heineman-Stiftung Foundation of West Germany and by Eastern Illinois University. The work was primarily done at von Karman Institute for Fluid Dynamics, Rhode-ST-Genese, Belgium.

1. The Objective.

Let us consider a nonlinear partial differential equation

$$\partial u/\partial t = L(u) \qquad (1.1)$$

where L is a one-dimensional differential operator in x. Let the domain of integration be $[a,b] \times [0,\infty)$. Equation (1.1) is subject to certain initial/boundary conditions and it is assumed that the problem is mathematically well-posed.

An explicit finite difference analog of (1.1) is

$$U^n = F(U^{n-1}) \qquad (1.2)$$

where, $U^n = (U_1^n\ U_2^n \ldots U_I^n)^T \varepsilon\ D \subset R^I$, ($R^I$ = the real I-dimensional space), $U_i^n = U(x_i,t_n)$ = the net function corresponding to u_i^n which is the true value of u at (x_i,t_n).

An implicit finite difference analog of (1.1) is:

$$G(U^n) = U^{n-1}. \qquad (1.3)$$

Also, $F: D \subset R^I \to D$ and so is G. It is assumed that the truncation errors are small and their effects are negligible.

Grid error is defined by

$$e^n = u^n - U^n. \qquad (1.4)$$

Stability is guaranteed iff $\forall\ n, \|e^n\| < K$, where K is positive and arbitrarily chosen.

In this article an attempt will be made to see how one can obtain, $\forall\ e^1\ \varepsilon\ R^I$

$$\lim_{n\to\infty} \|e^n\| = 0 \qquad (1.5)$$

for given Δx (mesh size) and Δt (time step).

Obviously (1.5) guarantees stability. It also implies convergence for steady state solution.

2. Mathematical Preliminaries.

Let, $\forall$ n, $z^n \in R^I$, and

$$z^n = A_n z^{n-1}. \tag{2.1}$$

Clearly, $\lim\limits_{n\to\infty} z^n = 0$ iff

$$\lim_{n\to\infty} A_n A_{n-1} \dots A_1 = 0. \tag{2.2}$$

Now (2.2) is true if there exists a particular norm such that $\forall$ $n > N$

$$\|A_n\| \le \alpha < 1. \tag{2.3}$$

(These are discussed in details in [2].) Under these conditions (2.1) is said to describe a motion of decadence and A_n is called a D-matrix.

If instead of (2.1) the motion is given by

$$A_n z^n = z^{n-1} \tag{2.4}$$

it is a motion of decadance iff A_n^{-1} is a D-matrix which is true if

$$\|A_n^{-1}\| \le \alpha < 1 \tag{2.5}$$

for some particular norm and $\forall$ $n > N$.

It may be proved:

Theorem: 1 If A_n is a lower triangular matrix and $\rho(A_n) \le \alpha < 1$, A_n is a D-matrix. ($\rho(A_n)$ = Spectral Radius of A_n.)

Theorem: 2 If A_n is a tridiagonal matrix and (i) for $i \neq j$, $|a^n_{ij}| < |a^n_{ii}|$ and (ii) $|a^n_{ii} - (a^n_{i,i-1}\, a^n_{i-1,i}/a^n_{i-1,i-1})| > 1$, $|a^n_{11}| > 1$, A_n^{-1} is a D-matrix. The same is true if A_n is a bidiagonal matrix with nonnull elements on the main diagonal.

3. Analysis of Discretization Errors.

Let us consider (1.2). Let

$$F(u^{n-1}) - F(U^{n-1}) = A_n e^{n-1}. \qquad (3.1)$$

Obviously, if a^n_{ij} is an element of A_n, $a^n_{ij} = a^n_{ij}(u^n, U^n)$. Then the grid error equation for (1.2) is:

$$e^n = A_n e^{n-1}. \qquad (3.2)$$

Hence, (1.5) is true if A_n is a D-matrix.

If we express,

$$G(u^n) - G(U^n) = A_n e^n \qquad (3.3)$$

then for (1.3), the equation (1.5) is true if A_n^{-1} is a D-matrix.

It may be seen that the effects of truncation error are totally neglected in this discussion. Such effects were discussed in [2].

Thus, for an explicit finite difference equation, grid error effects are damped out if A_n in (3.2) is a D-matrix; and for an implicit finite difference equation, the same is true if A_n in (3.3) is such that A_n^{-1} exists and is a D-matrix.

4. Algorithm for Stability Analysis.

It is well known that for any square matrix $A_n (I \times I)$

$$\max_{ij} |a^n_{ij}| \leq \|A_n\| \leq I \cdot \max_{ij} |a^n_{ij}| \tag{4.1}$$

for certain natural norms. Thus, for an explicit equation like (1.2), (1.5) is true if

$$I \cdot \max_{ij} |a^n_{ij}| \leq \alpha < 1. \tag{4.2}$$

If in case A_n is a lower triangular matrix, Theorem: 1 may be applied.

For an implicit equation of the form (1.3), if A_n is a tridiagonal matrix, grid error effects may be studied by using Theorem: 2. A general analysis for A_n (or A_n^{-1}) to be a D-matrix may be found in [5].

5. Application.

Let us consider the inviscid Burgers' equation:

$$u_t + (1/2)(u^2)_x = 0. \tag{5.1}$$

Let the initial conditions be:

$$\begin{aligned} u(x,0) &= V_1 \text{ if } x \leq x_J \\ &= V_2 \text{ if } x > x_J, \\ V_1 &> V_2 \end{aligned} \tag{5.2}$$

Let u_t be approximated by a two-point forward difference formula and $(u^2)_x$ be approximated by a two-point backward difference formula. Then the difference approximation of (5.1) is:

$$u_i^{n+1} = a(u^n_{i-1})^2 - a(u^n_i)^2 + u^n_i + \tau^n_i$$

If u^n_i is replaced by U^n_i and τ^n_i (the truncation error) is dropped, then using $e_i^n = u_i^n - U_i^n$, we get:

$$e_i^{n+1} = a(u_{i-1}^n + U_{i-1}^n)e_{i-1}^n + \{1 - a(u_i^n + U_i^n)\}e_i^n \qquad (5.3)$$

where $a = \Delta t/(2\Delta x)$.

The linearized stability analysis requires:

$$a(2V_1) \le 1 \qquad (5.4)$$

where $V_1 = \max\limits_{i,n} |u_i^n|$. This inequality implies restriction on time step given by:

$$\Delta t \le \Delta x/V_1. \qquad (5.5)$$

In the present analysis (5.3) may be expressed as:

$$e^{n+1} = A_n e^n \qquad (5.6)$$

where A_n is a bidiagonal matrix having diagonal elements $a_{ii}^n = 1 - a(u_i^n + U_i^n)$ and elements below the main diagonal as $a_{i,i-1}^n = a(u_{i-1}^n + U_{i+1}^n)$. Then by Theorem: 1, A_n is a D-matrix if

$$\max_i |a_{ii}^n| \le \alpha < 1, \qquad \forall\, n > N. \qquad (5.7)$$

If one chooses arbitrarily $V_1 = 1.3$, $V_2 = 0.0$, $\Delta t = \Delta x = 0.1$ (and $x_J = x_4$), the linearized stability criterion (5.5) is violated, although (5.7) is satisfied. Computationally, instabilities were not found and the results given by fig. 1 seem to be quite correct.

Stability analysis of other explicit finite difference analogs may be treated similarly or by using the inequality (4.1).

If both u_t and $(u^2)_x$ are approximated by two point backward difference formulas, we get an implicit finite difference analog of (5.1) and dropping the truncation error, the grid error equation becomes:

$$-a(u^n_{i-1} + U^n_{i-1})e^n_{i-1} + \{1 + a(u^n_i + U^n_i)\}e^n_i = e^{n-1}_i. \qquad (5.8)$$

Here, A_n is a diagonal dominant lower triangular matrix and $|a^n_{ii}| > 1 \;\forall\; n > N$. Hence, the numerical scheme is unconditionally stable by Theorem: 2.

Let (5.1) be expressed as:

$$u_t + uu_x = 0. \qquad (5.9)$$

If u_t is approximated by a two-point backward difference formula and u_x is approximated by a central difference formula, the error equation becomes:

$$-aU^n_i e^n_{i-1} + \{1 + a(u^n_{i+1} - u^n_{i-1})\}e^n_i + aU^n_i e^n_{i+1} = e^{n-1}_i. \qquad (5.10)$$

Here, A_n is a tridiagonal matrix and considering the initial conditions (5.2), $|a^n_{ii}| \ngtr 1$. Hence, Theorem: 2 cannot be applied. Thus, stability criterion is not satisfied. (Linearized stability criterion is, however, unconditionally satisfied.) Actual computations showed instabilities. Now if we change the initial boundary conditions as: $u(x,0) = x$, $u(0,t) = 0$, $u(1,t) = 1/(1 + t)$, $u_{i+1} > u_{i-1} \;\forall\; i$ and $|a^n_{ii}| > 1$ with diagonal dominance, the implicit scheme should now be unconditionally stable. Ziebarth [6] verified it computationally.

6. A Remark on Navier-Stokes' Equation.

Let us consider Navier-Stokes' equation in the vorticity-stream function form as:

$$\zeta_t + \zeta_x\psi_y - \zeta_y\psi_x = \nu\nabla^2\zeta \qquad (6.1)$$

$$\nabla^2\psi = -\zeta \qquad (6.2)$$

where ζ = vorticity and ψ = stream function. This coupled system is subject to some specified initial-boundary conditions. If we analyze the grid errors for implicit schemes we get two equations of the form

$$\phi_n e^n + \theta_n f^n = e^{n-1} \qquad (6.3)$$

$$\Lambda_n f^n = e^n \qquad (6.4)$$

where e^n = grid error for ζ and f^n = grid error for ψ [7]. These equations may be expressed as

$$A_n e^n = e^{n-1}. \qquad (6.5)$$

It appears that if sharp discontinuities are present neither in the flow field nor on the boundary, conditions of Theorem: 2 will be satisfied. Therefore, the implicit scheme will be stable.

7. Conclusion.

If the sequence of matrices $\{\Lambda_n\}$ be such that $\forall$ n, $\|\Lambda_n\| \leq \alpha < 1$, $\|e^n\|$ will form a monotone decreasing sequence, whereas if $\forall$ $n > N$, $\|\Lambda_n\| \leq \alpha < 1$, $\|e^n\|$ may show some oscillations before it is damped out. In both cases, however, as $n \to \infty$ $\|e^n\| \to 0$.

For the linearized grid error analysis, $\Lambda_n = \Lambda$ $\forall$ n and if Λ is a convergent matrix stability is obtained. Thus, linearized grid-error theory is a particular case of the analysis presented here.

Since elements of A_n are functions of u^n and U^n, initial-boundary conditions affect the properties of A_n.

In order to check that A_n (or A_n^{-1}) is a D-matrix, some information regarding the nature of the solution must be known a priori. This may be done mathematically or experimentally or both.

References

1. S. K. Dey: Nonlinear Discretization Errors in Partial Difference Equations. BIT, Vol. 20, No. 1, 1980.

2. S. K. Dey: Numerical Instabilities of Nonlinear Partial Differential Equations. CFT 7800/SKD/IE. von Karman Institute for Fluid Dynamics, Rhode-ST-Genese, Belgium, 1978.

3. W. F. Ames: Numerical Methods for Partial Differential Equations, Barnes and Nobles, Inc. New York, 1969.

4. R. D. Richtmyer and K. W. Morton: Difference Methods for Initial-Value Problems. Interscience Publishers. New York, 1967.

5. S. K. Dey: Analysis and Applications of Mappings of Decadence. Eastern Illinois University.

6. J. Ziebarth: Comparative Studies of the Computational Analysis of One Dimensional Gas Flow. Masters Thesis. Department of Mathematics, Eastern Illinois University, 1975.

7. S. K. Dey: Error Propagation on Implicit Finite Difference Solution of Navier-Stokes' Equation. ZAMM. (To be published.)

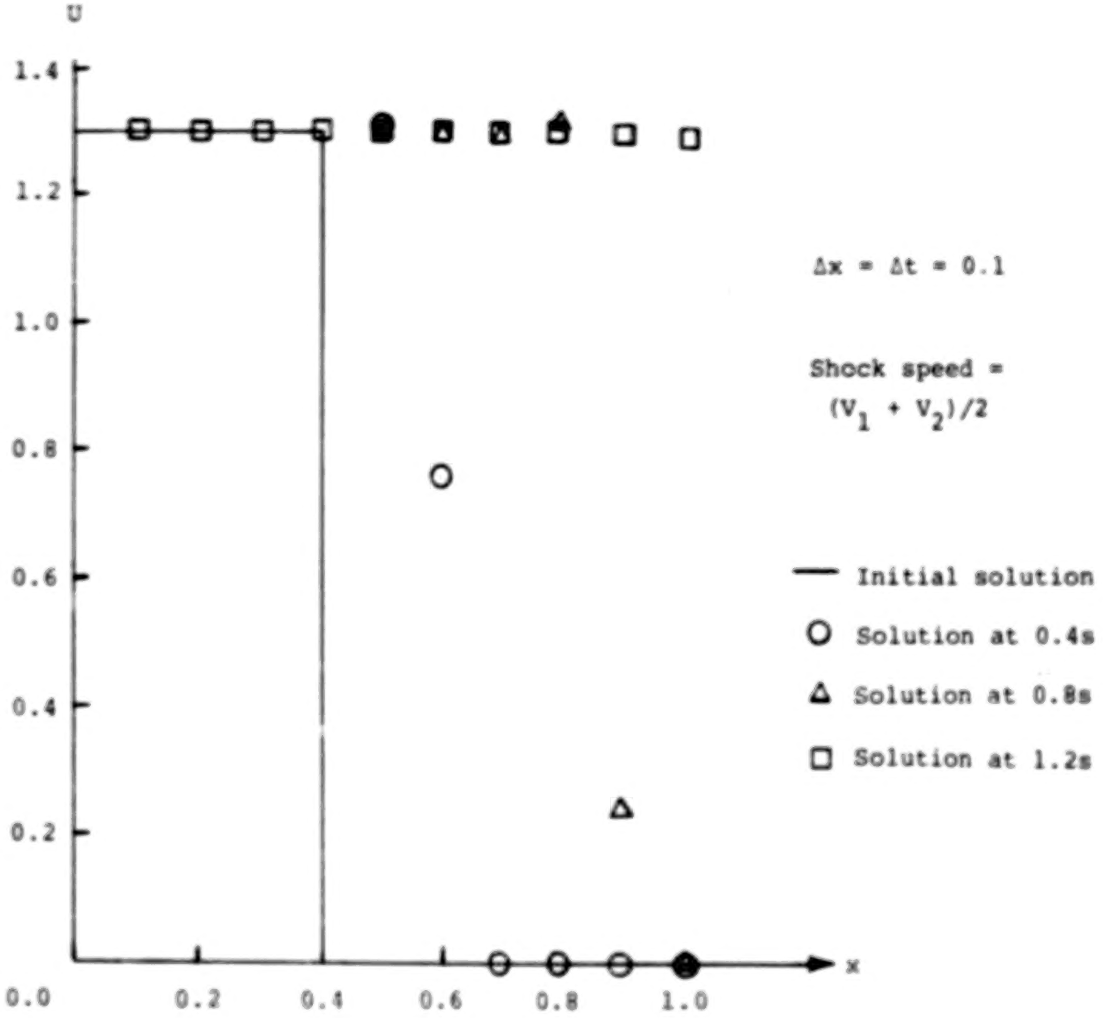


Figure 1.- Explicit finite difference solution of equation (5.1).

GRID GENERATION
USING
COARSE, SMOOTH FINITE ELEMENTS

Lawrence J. Dickson
Department of Aeronautics
and Astronautics
University of Washington

I. Introduction

The grid generation problem lends itself to the use of finite elements and variational equations.

(1) Grids are usually generated as smooth solutions to "nice," elliptic differential equations--just the equations well suited to variational methods.

(2) The use of smooth finite elements gives the grid a functional expression, which can be examined, evaluated, manipulated, and modified naturally and cheaply.

(3) The "grid equations" are chosen for their qualitative character. Exactitude of solutions does not matter as long as this is preserved. As a result, extremely coarse (cheap) finite elements may generate a grid of high quality, if the boundary conditions are well parametrized.

I succeeded in demonstrating the following:

(1) Grid-quality solutions of a wide variety of equations--(direct) Laplace's, biharmonic, Helmholtz, even nonlinear--can be generated to fit reasonable functional boundary conditions in 2D using very coarse rectangular finite elements, often 6x3 C^2 bicubic. I even tried some "wavy" operators (with no natural variational expression) to demonstrate the method's versatility. I did not try the inverse Laplace equation, but I expect no problem but cost.

(2) The finite element grids can be refined, locally modified and "fine-tuned" using a simple, cheap composition-of-functions approach, without having to solve the differential equation repeatedly.

II. The Finite Elements

Smooth, rectangular finite elements were used; the ones here are C^2 bicubic in the interior. For the linear equations, an option of C^1 cubic boundary conditions with arbitrarily dense nodes was included. In the examples shown there are, unless otherwise mentioned, six patches "circumferentially" and three "radially," of which only the innermost ring of patches "radially" is plotted.

Where C^1 boundary conditions are used, second derivative discontinuity in the "circumferential" direction is confined to the ring(s) of elements meeting the C^1 boundary.

III. The Variational Expressions

Both linear and non-linear equations are solved by minimizing a variational integral. In the non-linear case there is iteration. Equations to fourth order (i.e., expressions squared in the variational integral to second order) are treated. In the case of the "wavy" Helmholtz equation $H(f) = \nabla^2 f + k^2 f = 0$, the variation integral for $H*H$ is used with Dirichlet ("underdetermined") boundary conditions.

A variational approach to solving the inverse Laplace's equation is known, but was not tried in this research. Other inverse equations (such as biharmonic) could also be used.

Equation	Variational Integrand
$\nabla^2 f = 0$	$\lvert\nabla f\rvert^2$
$\nabla^2 f - k^2 f = 0$	$\lvert\nabla f\rvert^2 + k^2 f^2$
$\nabla^2 f + k^2 f = 0$	$(\nabla^2 f + k^2 f)^2$
$\nabla^4 f = 0$	$(\nabla^2 f)^2$
$\nabla^2 (f^2) = 0$	$\lvert f\nabla f\rvert^2$

The boundary conditions are always "Dirichlet-like" (Dirichlet for second order, Dirichlet or Dirichlet + Neumann for 4th order), which allows simple minimization of the integral.

Since the finite element grids were so coarse, solution of the equations was by a one-step symmetric matrix solver.

IV. Fine-Tuning

The grids are defined by functional equations $x = x(i,j)$, i and j being "counting" variables. But the solutions to the differential equations are $x = x(s,t)$. It remains to define $s(i,j)$ and $t(i,j)$.

The simplest approach is to make them merely linear functions. As a matter of fact, I imposed the boundary conditions $s(i_L,j) = \text{const}$, $s(i_R,j) = \text{const}$, $t(i,j_B) = \text{const}$, $t(i,j_T) = \text{const}$, so that (s,t) lies in a box; then I manipulated the interior values through transfinite interpolation. If i is "circumferential" the maps are as follows:

$$\begin{tikzcd}
\binom{i}{j} \arrow[r, "\begin{array}{c} i = u \\ j = c(u,v) \\ \text{(unbroken cubic in } v\text{)} \end{array}"'] & \binom{u}{v} \arrow[r, "\text{TFI}"'] & \binom{s}{t} \arrow[r, "\text{FE}"'] & \binom{x}{y}
\end{tikzcd}$$

Where the F.E. solution has high skewness or is nearly singular, the first map allows the requirement on the TFI boundary condition needed to mend this to be multiplied by a small constant, avoiding "grid folding." A price is paid; the grid comes out looking irregular. It is better to avoid the skewness in the FE solution itself, as by using the biharmonic equation with perpendicular boundary conditions.

V. Conclusions

The finite element approach to grid generation has proved eminently successful for linear grid equations. Its coarseness is a very desirable characteristic, most encouraging from the point of view of extending it to 3D. The wide choice of equations--controlled by a small "black box" determining the variational integrand--is another asset. So is the method's usability, at little extra cost, with boundary conditions more finely specified than the interior finite element grid.

Iterative solution of nonlinear equations increases the expense by more than an order of magnitude, usually requires numerical integration of the variational expression (often convenient in the linear cases too), and makes it very difficult in general to apply finely-specified boundary conditions. I think there are better approaches (see below).

The "fine-tuning" needs further refinement itself, especially in handling variations in the normal velocity of the grid. Choosing the grid equation to yield perpendicularity of the finite element grid map (possible, for instance, with the biharmonic equation) is a help. Control of "circumferential" grid density at chosen locations worked well.

VI. Future Directions

Algorithms should be derived to parametrize boundary conditions in such a way as to yield good grids using direct (linear) equations, such as Laplace's or the biharmonic. The idea is to imitate the Riemann mapping, by slowing down where convex (avoiding boundary overlap), and speeding up where concave (avoiding "folds" in grid interior). This should make use of inverse equations less necessary, and if done right should be extendable to 3D.

The "fine-tuning" algorithms must be refined. They are in principle applicable to any grid that can be described as a function.

For the inverse equations (for instance, Thompson's method), I suggest use of linear paneling schemes in (x,y) space. The resulting solution can be approximated by (s,t) finite elements simply by solving for the (few) internal nodal values of (s,t) and using inverse function theorem derivative evaluations. If the solution needs to be iterated, linearized variational expressions using this as a starting point should be cheap. Such a method might even have uses in linearized flow simulation, as for cheap streamline tracking.

The method needs to be extended to 3D, adapted for "block" grids (with the equations, if desired, being valid across block boundaries), and adapted for vector computers.

VII. Figures

The figures are true representations of the functions they illustrate, although some "handwork" was done on some of them to circumvent bugs in the evaluation and plotting software.

All internal finite element grids are six circumferentially by three radially, with only the innermost radial layer plotted, unless otherwise mentioned.

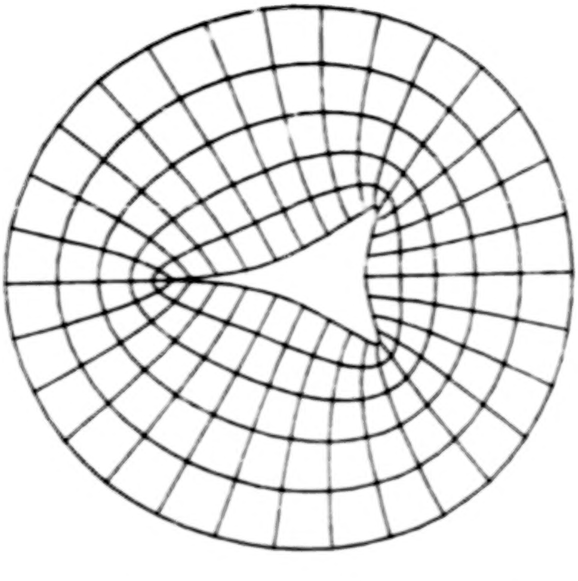


Figure 1.- "Heart" - Laplace's equation was approximated, with boundary conditions hand-parametrized to give a well-conditioned if not unskewed grid.

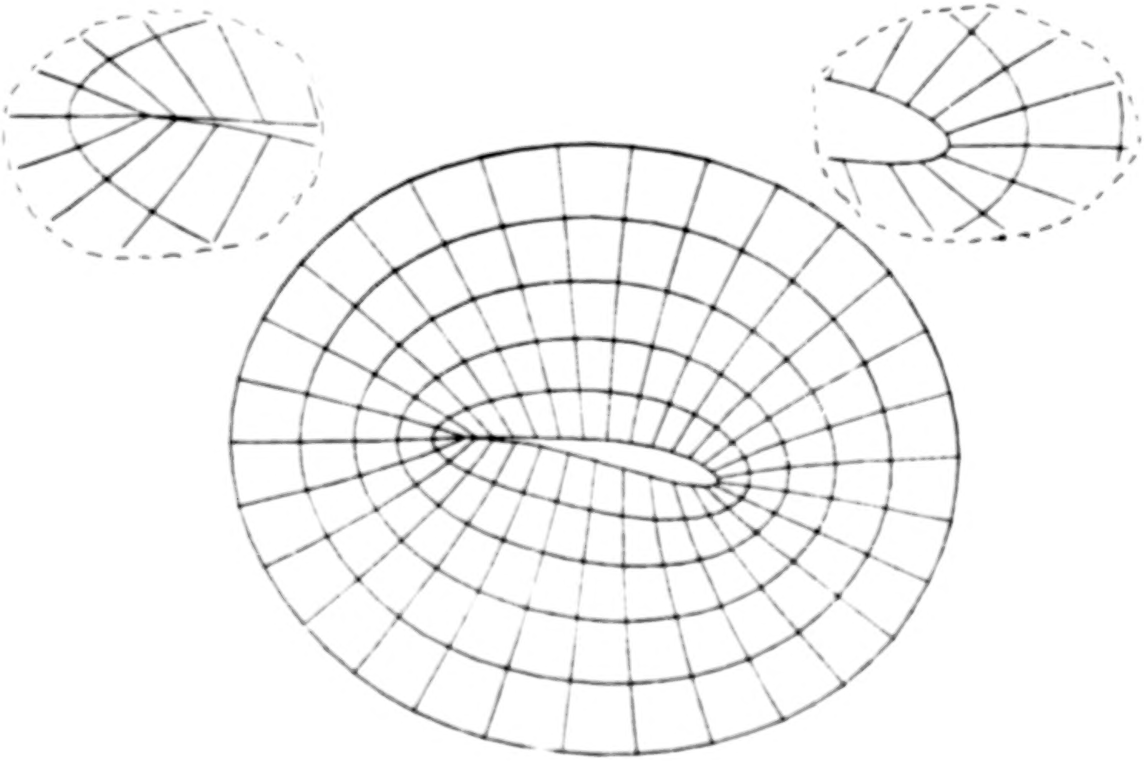


Figure 2.- "Joukowski" (Laplace's) - The natural, analytic parametrization of a Joukowski airfoil was imitated by a six-node C^2 cubic periodic spline. Success in avoiding skewness was middling, as the insets show. "Radial" velocity at trailing edge was non-zero.

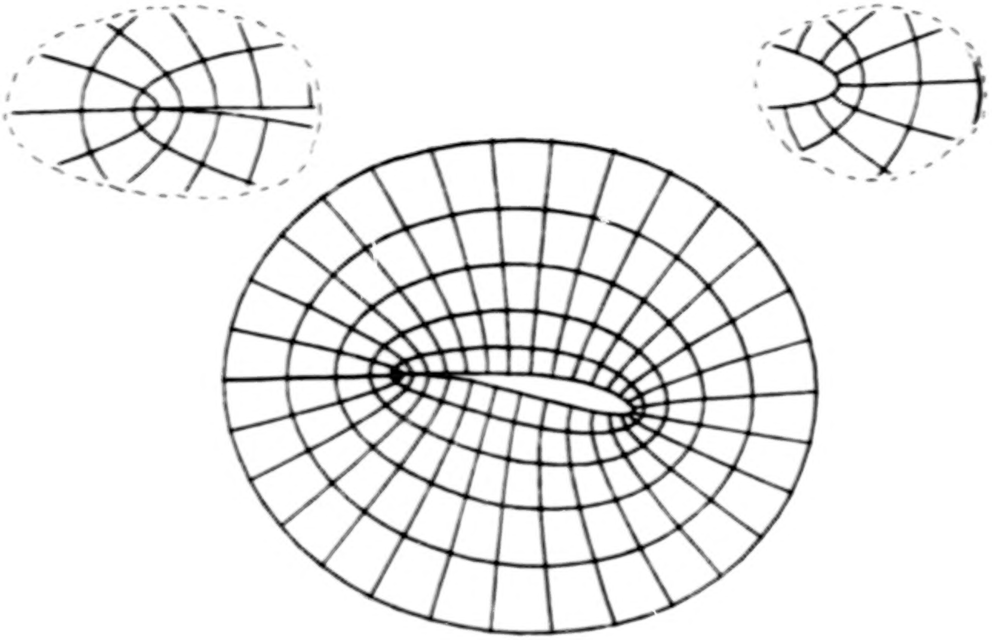


Figure 3.- "Joukowski" (Biharmonic) - The biharmonic equation's normal derivative condition was used to enforce conformality in the limit exactly at boundaries. (This makes the grid C^1 near the airfoil, and requires the "fine" boundary condition algorithm.) The insets show its success, and also that "normal" velocity at TE is zero.

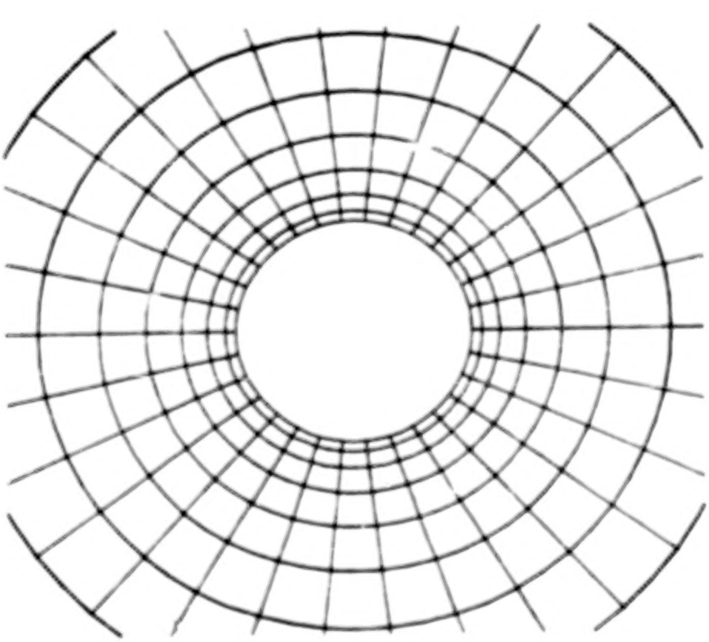


Figure 4.- "Helmholtz" - The equation is $\nabla^2 f - k^2 f = 0$, $k = .65$, with $0 \le s = 3\theta/\pi \le 6$ and $0 \le t \le 3$, radially symmetric boundary conditions, $r(t=0) = 1$, $r(t=3) = \exp(\pi)$. $0 \le t \le 1.4$ is plotted.

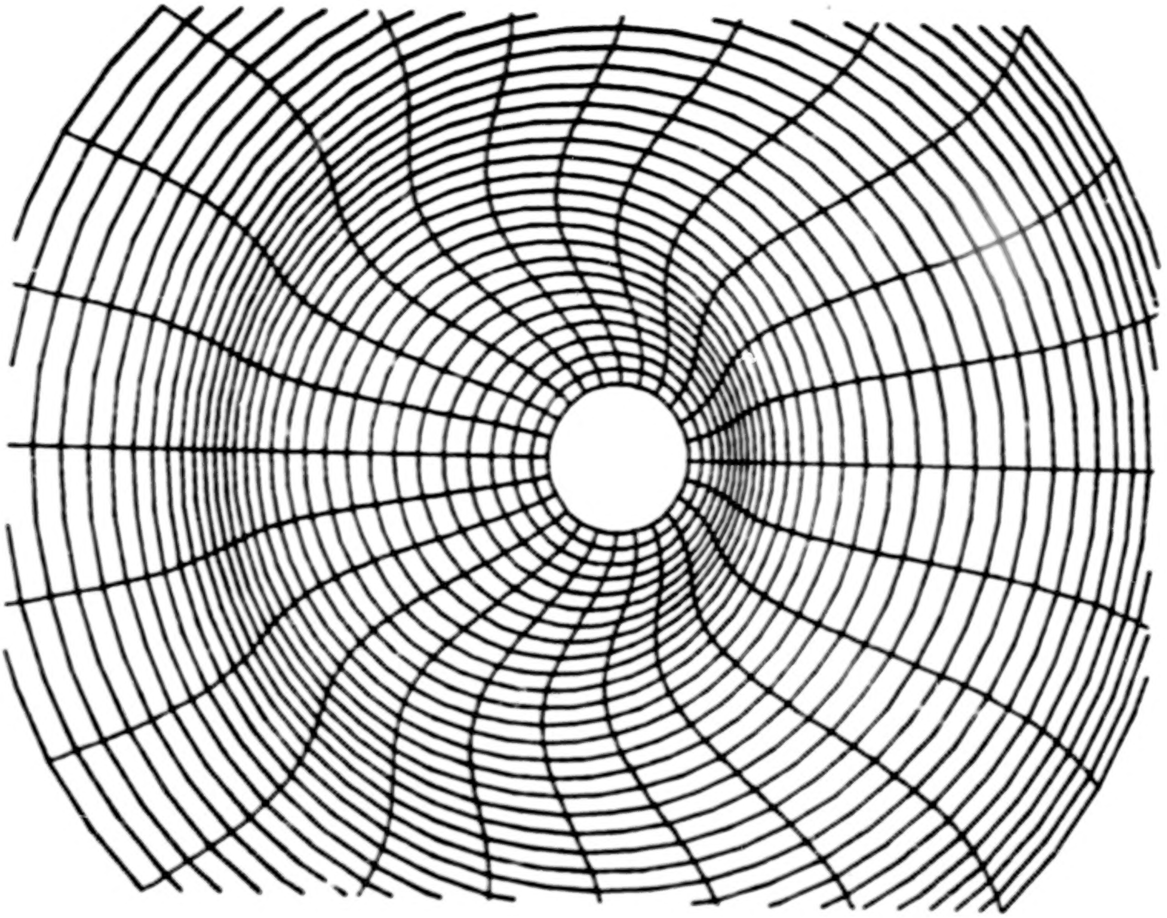


Figure 5.- "Wavy Helmholtz" - The equation is $^2f + k^2f = 0$, $k = /3$, with $0 \le s = 3/ \le 6$ and $0 \le t \le 6$. At $t = 0$, $(x,y) = (.5+\cos ,\sin)$, and at $t = 6$, $(x,y) = \exp( ) (\cos ,\sin)$. x includes a wave that traverses 1.25 cycle in joining these boundary conditions. The plot shows all the grid (except for screen cutoffs), which was solved on a 6·6 finite element mesh.

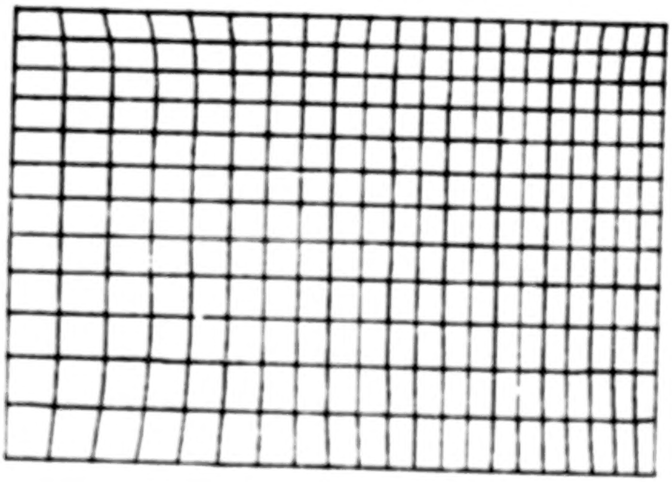


Figure 6.- Nonlinear - The equation is $^2(f^2) = 0$. This 3·3 grid, shown in its entirety, approximates $x = \sqrt{1.75t+1}$ and $y = \sqrt{s+1}$, as expected from equation and boundary conditions. Slight deviation is visible in x, due to the coarseness of the grid and the (2·2) Gaussian integration.

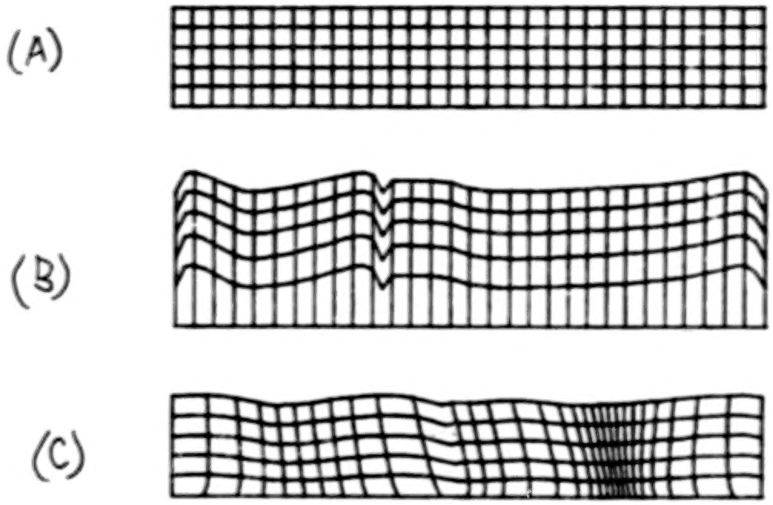


Figure 7.- "Fine tuning," Laplace - (A) = (i,j), (B) = (u,v), (C) = (s,t) in the discussion of Section IV. The wiggles in (B), due to skewness, are present only to second order in (C).

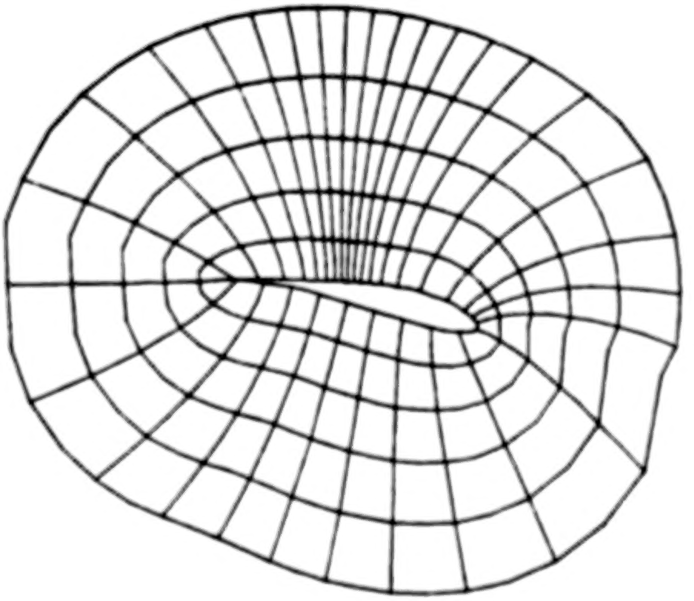


Figure 8.- "Fine tuned" Joukowski (Laplace) - The result of composition by the maps of Figure 7 is shown. (The "fine-tuning" specifications were deliberately clumsy.) The "shock densing" is good and the TE not too bad, but the LE shows a nasty glitch where skewness had to be corrected. The finite element map is that of Figure 2.

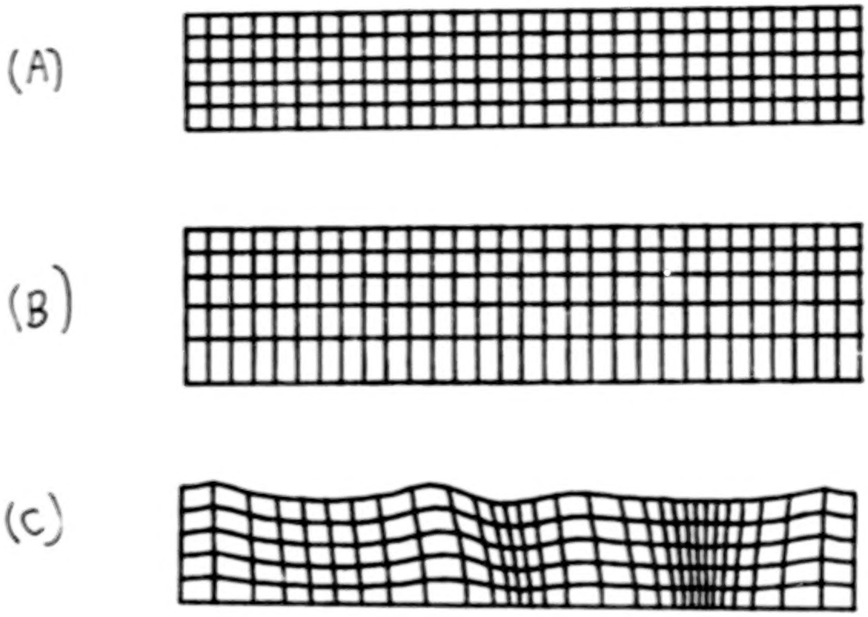


Figure 9.- "Fine tuning," Biharmonic - Smoother than Figure 7, since there is no skewness to be corrected at the boundary.

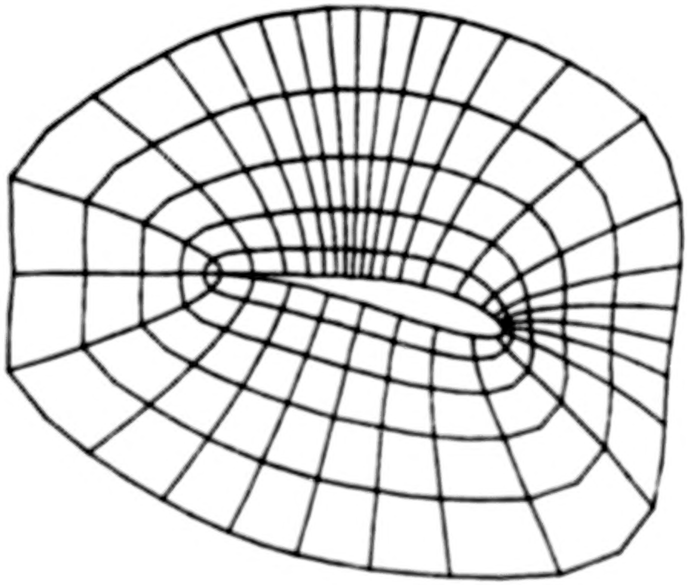


Figure 10.- "Fine tuned" Joukowski (Biharmonic) - Much better than Figure 8 at the LE, due to lack of skewness at the boundary in the map of Figure 3. The TE is not so good, due to adjusting to zero "normal" velocity there, a defect more easily correctable than the problem in Figure 8.

FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS

Djordje S. Dulikravich[*]
NASA Lewis Research Center

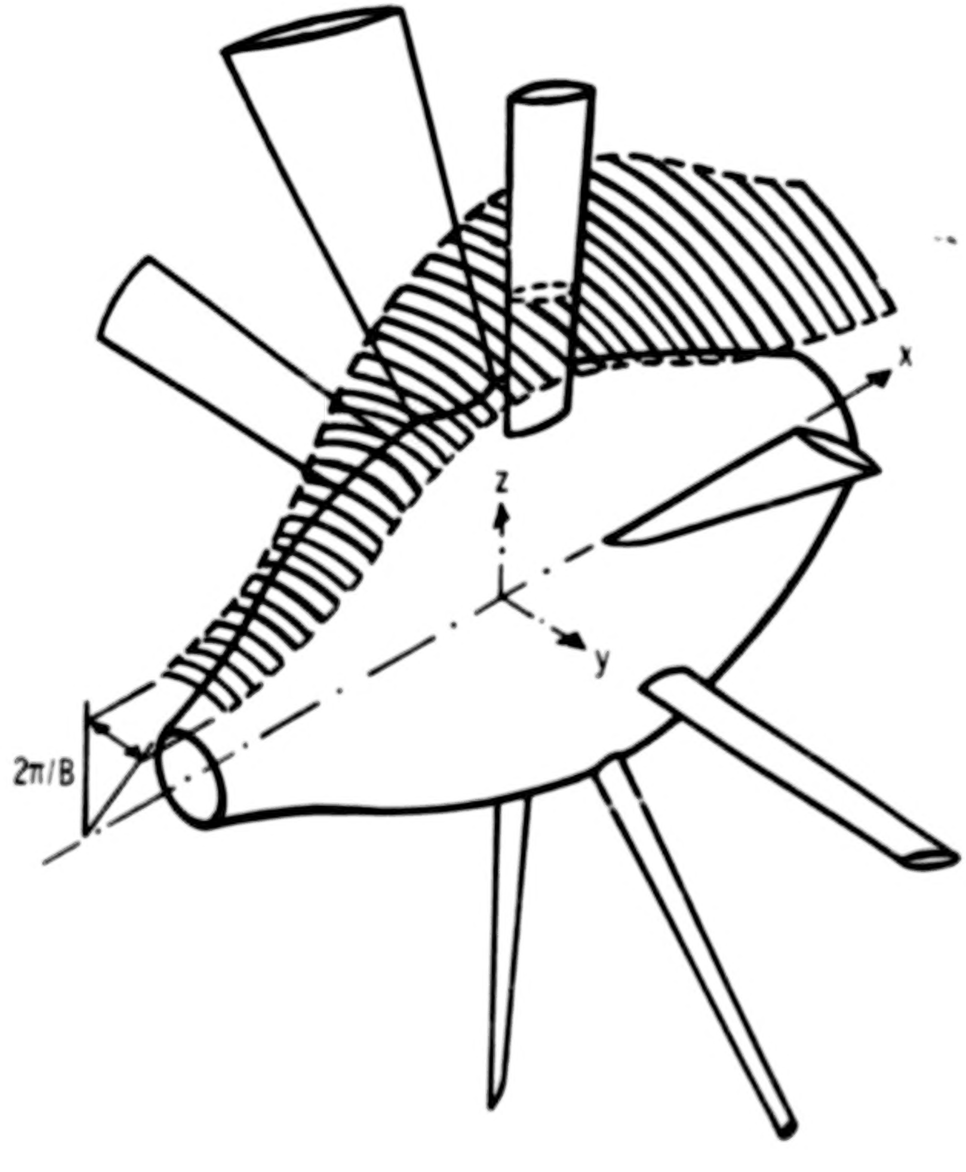
A fast algorithm has been developed for accurately generating boundary conforming, three-dimensional, consecutively refined, computational grids applicable to arbitrary axial turbomachinery geometry. The method is based on using a single analytic function to generate two-dimensional grids on a number of coaxial axisymmetric surfaces positioned between the hub and the shroud. These grids are of the "O"-type and are characterized by quasi-orthogonality, geometric periodicity, and an adequate resolution throughout the flowfield. Due to the built-in additional nonorthogonal coordinate stretching and shearing, the grid lines leaving the trailing edge of the blade end at downstream infinity, thus simplifying the numerical treatment of the three-dimensional trailing vortex sheet.

---

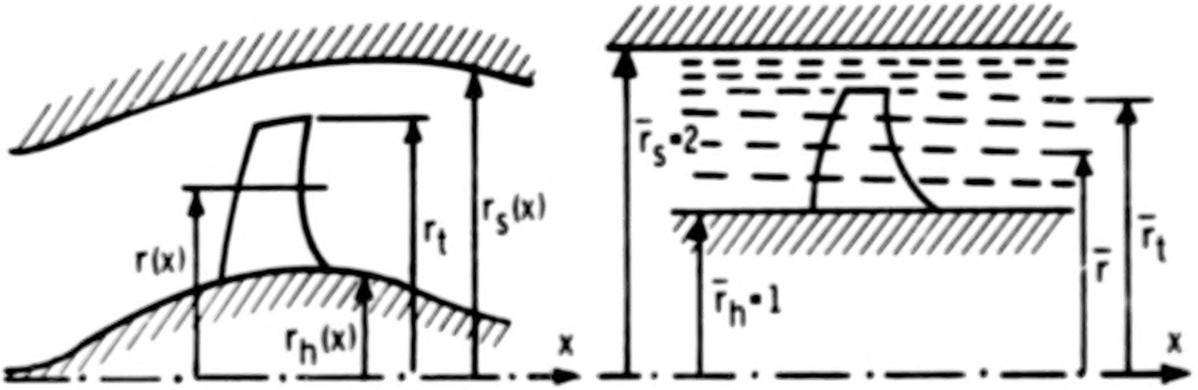
[*] NRC-NASA Research Associate, now a visiting research scientist at DFVLR-Göttingen Universität, F.R. Germany.

The main objective of this work is to develop a fast algorithm for generating body-conforming three-dimensional computational grids. An equally important objective is to preserve the high accuracy of the discretized representation of the solid boundaries. When analyzing steady flows through turbomachinery rotors and stators, it is sufficient to consider a single rotationally periodic segment of the flowfield. This segment is a doubly infinite strip stretching in the direction of the axis of rotation. The strip has a constant angular width of $2\pi/B$ where B is the total number of blades. Each blade has an arbitrary spanwise distribution of taper, sweep, dihedral and twist angle. The local airfoil shapes can vary in an arbitrary fashion along the blade span. The rotor hub and the duct (or shroud) can have different arbitrary axisymmetric shapes.

Such an arbitrary three-dimensional physical domain (Fig. 1) is first discretized in the spanwise direction by a number of coaxial axisymmetric surfaces which are irregularly spaced between hub and shroud.

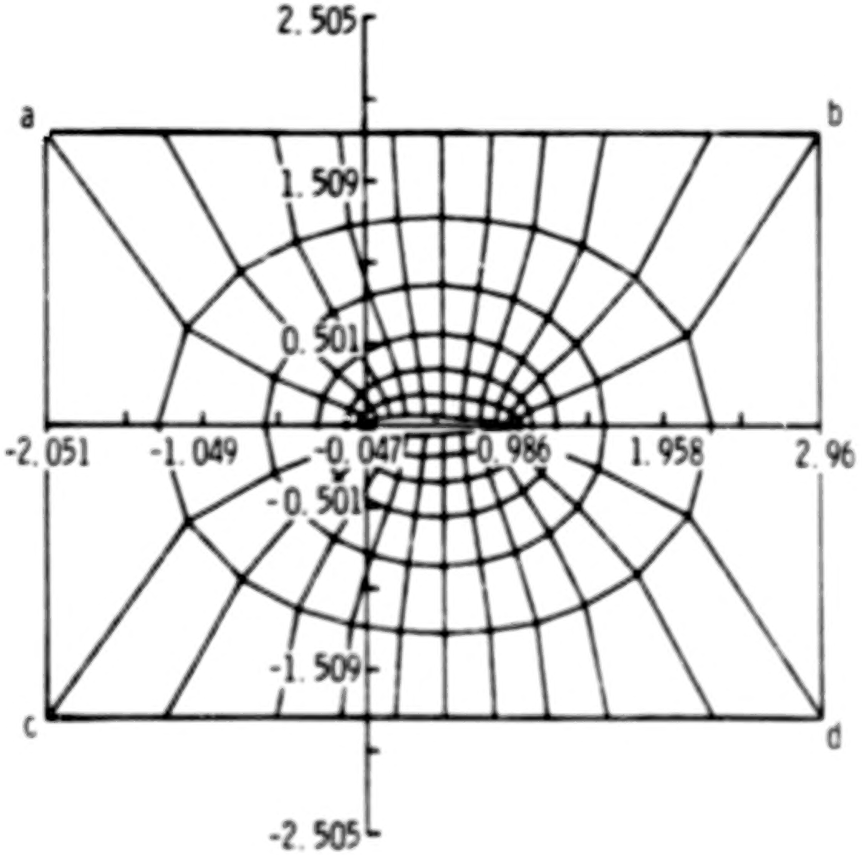


The major problem in generating the spanwise surfaces is an accurate determination of the intersection contours between the irregular blade surface and the coaxial axisymmetric surfaces cutting the blade. The coordinates of the points on these contours are defined by fitting cubic splines along the blade and interpolating at the radial stations corresponding to each axisymmetric surface $\bar{r}$ = Constant.

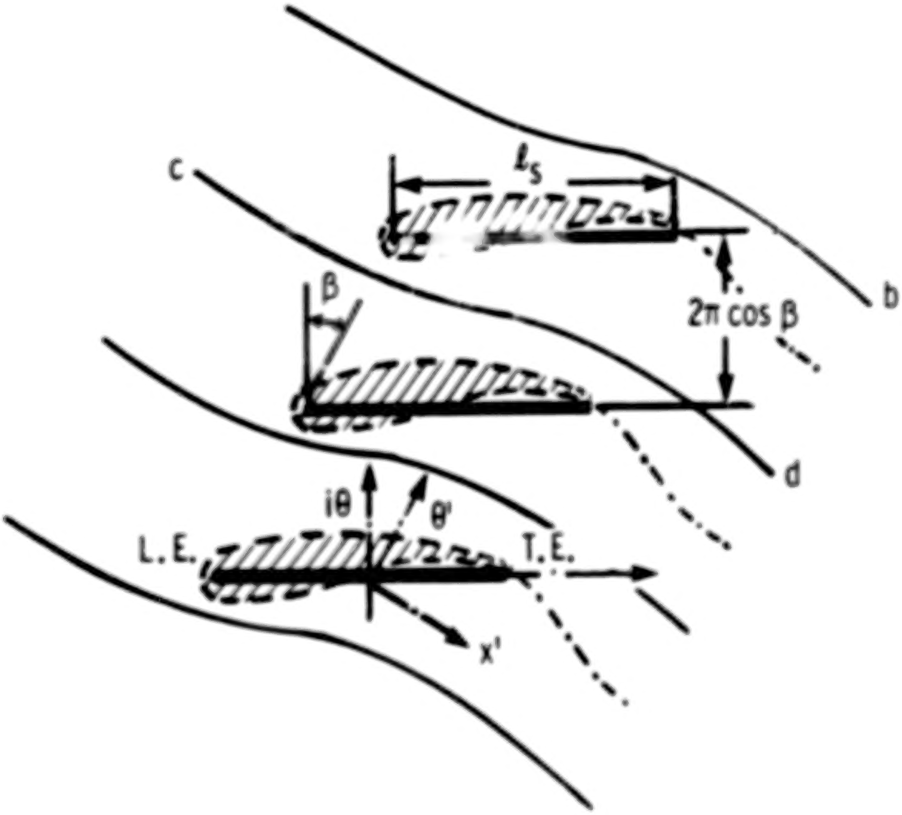


$$\bar{r} = 1 + \frac{r(x) - r_h(x)}{r_s(x) - r_h(x)}$$

The two-dimensional grid should have the following features: (a) grid cells should conform with the contour shape and the shape of the periodic boundaries $\overline{ab}$ and $\overline{cd}$, (b) grid should be geometrically periodic in the θ'-direction meaning that the grid points along the periodic boundary $\overline{ab}$ must have the same respective x'-coordinates as the grid points along the periodic boundary $\overline{cd}$, (c) grid lines should not be excessively non-orthogonal in the vicinity of solid boundaries, (d) a grid line emanating from the trailing edge should end at downstream infinity and (e) grid cells should be concentrated in the regions of high flow gradients.



Once the shape of the intersection contour on a particular cutting axisymmetric surface is known, the problem becomes one of discretizing a doubly connected two-dimensional domain $\tilde{w} = x + i\theta$.



ℓ_s
c
b
β
2π cos β
d
iθ
θ'
L.E.
T.E.
x'

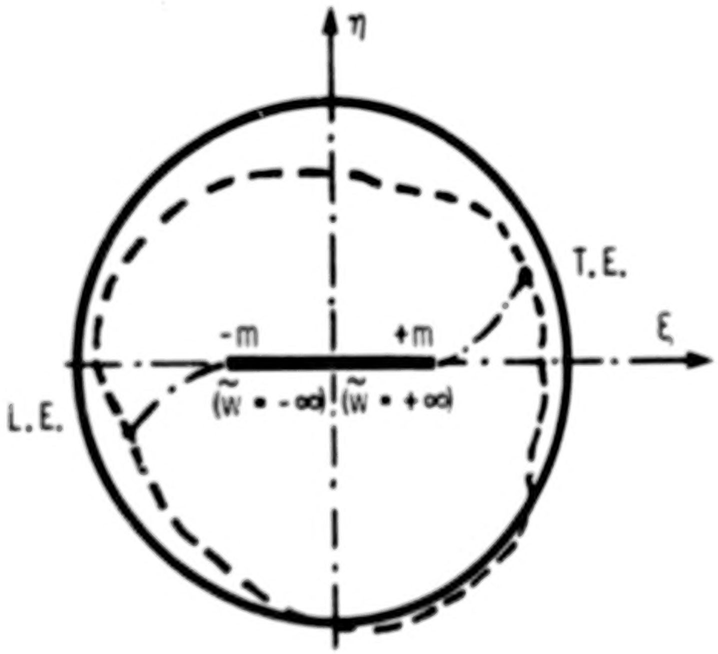
A grid with these properties can be most easily generated with the use of a single analytic function. One such function is

$$\tilde{w} = e^{i\beta} \ln\left(\frac{m - \tilde{z}}{m + \tilde{z}}\right) + e^{-i\beta} \ln\left(\frac{1 - m\tilde{z}}{1 + m\tilde{z}}\right); \quad 0 < m < 1$$

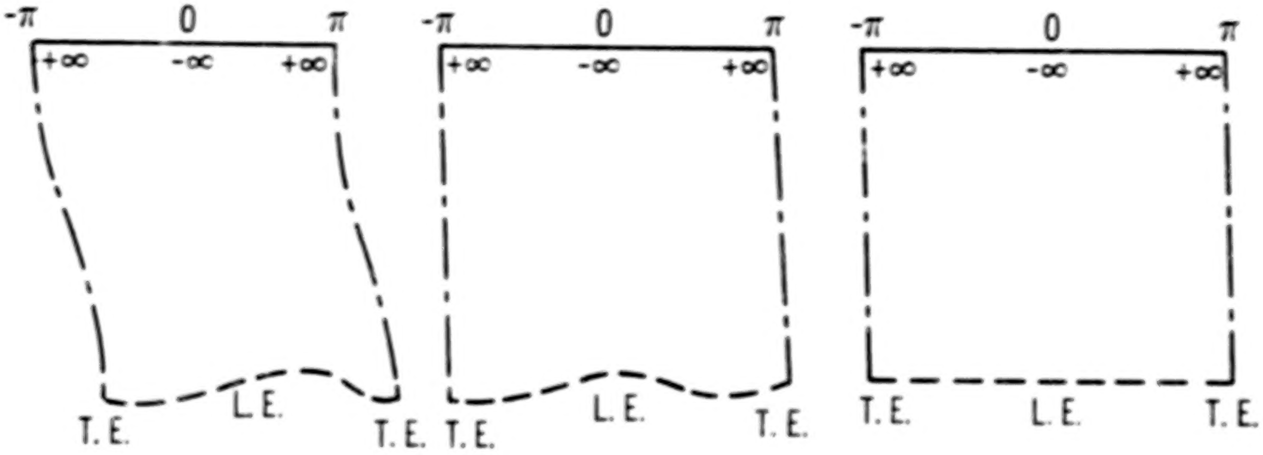
where $\tilde{w} = x + i\theta$ and $z = \xi + i\eta$. This complex function maps conformally a unit circle with a slit in the middle whose end-points are situated at $\tilde{z} = \pm m$ onto the cascade of straight slits in the $\tilde{w}$-plane. Each slit has a length ℓ_s where

$$\ell_s = 4\left(\cos\beta \sinh^{-1} \frac{2m \cos\beta}{1 - m^2} + \sin\beta \sin^{-1} \frac{2m \sin\beta}{1 + m^2}\right)$$

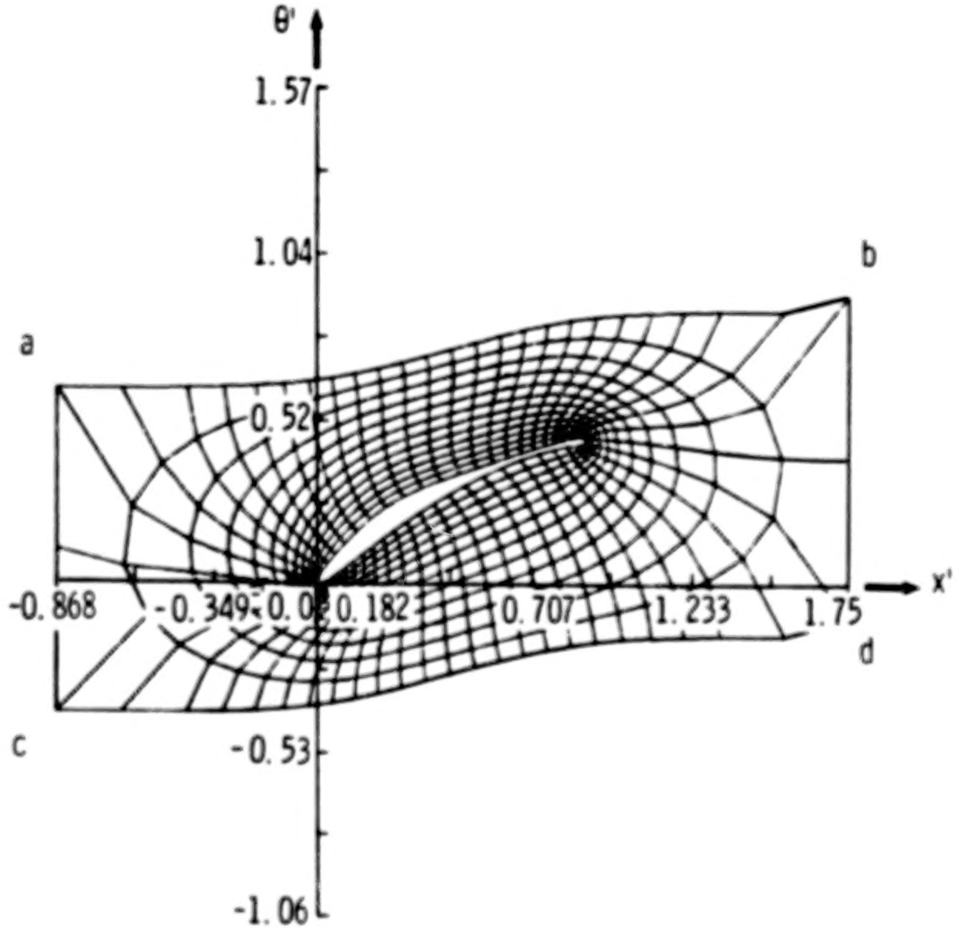
The slits are spaced $2\pi \cos\beta$ distance apart, where β is the stagger angle of the cascade of slits.



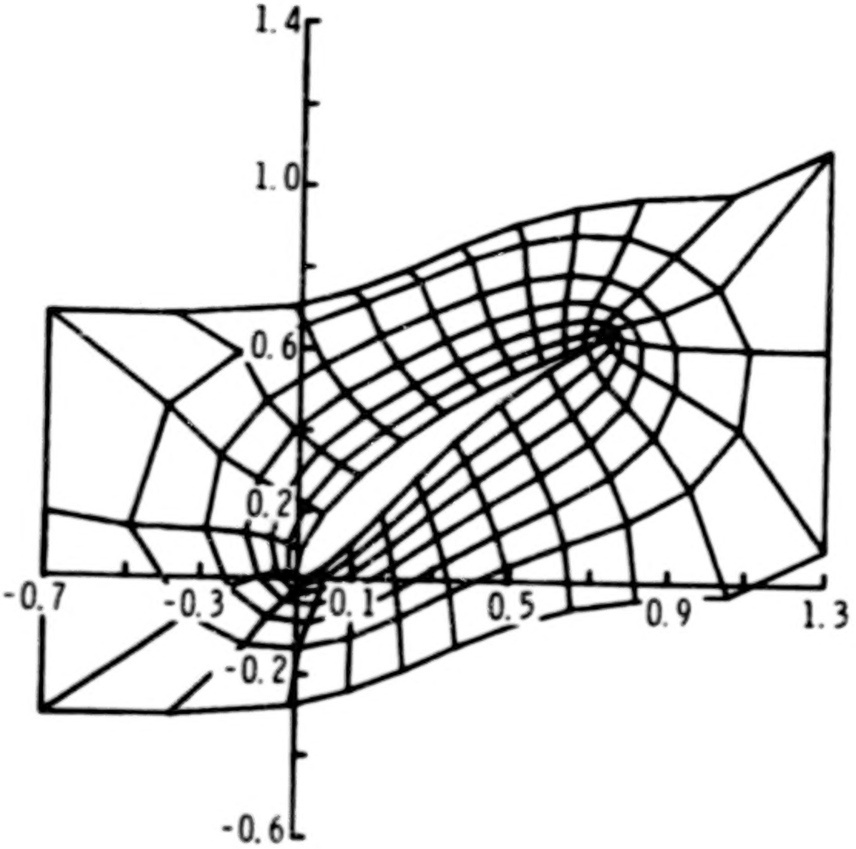
The unit circle is "unwrapped" using elliptic polar coordinates (refs. 1 and 2) resulting in a deformed rhomboidal shape which is then sheared in the horizontal and vertical direction (ref. 2) resulting in a rectangular (X,Y) computational domain.


-π
0
π
+∞
-∞
+∞
T.E.
L.E.
T.E.
-π
0
π
+∞
-∞
+∞
T.E.
L.E.
T.E.
-π
0
π
+∞
-∞
+∞
T.E.
L.E.
T.E.

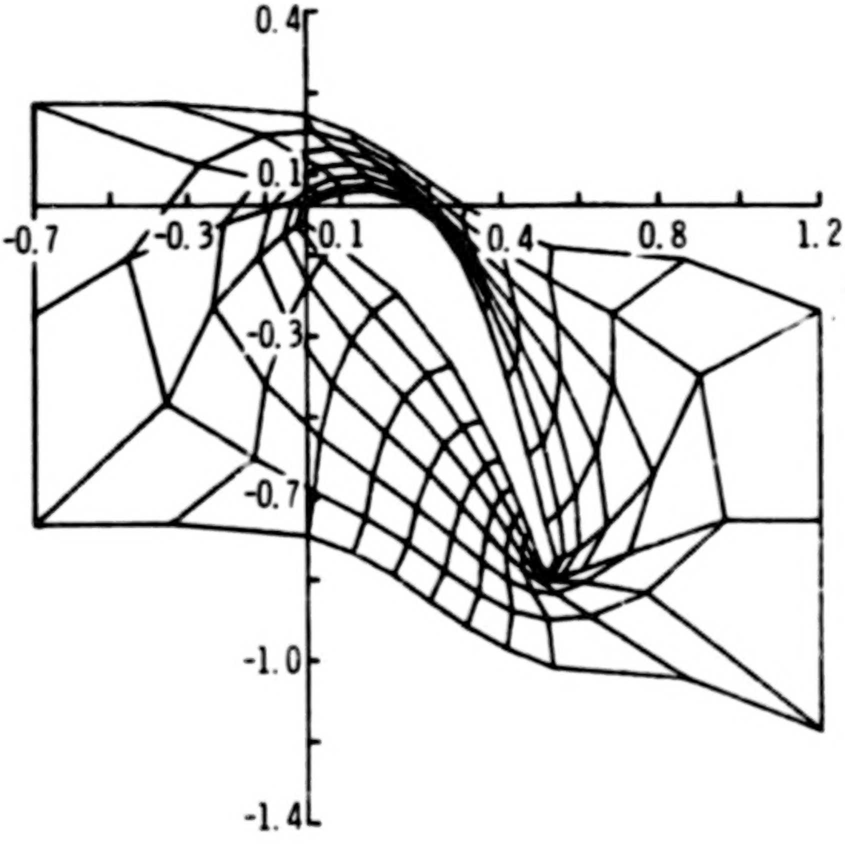
The transformation of an actual cascade of airfoils will result in a cascade of unit circles which are even more deformed. Consequently, more nonorthogonality will be introduced in the transformation by additional shearing of coordinates. A uniform grid in the (X,Y) plane which is symmetrically spaced with respect to the Y-axis, remaps back into the physical (x,θ) plane as an "O"-type boundary conforming grid. The actual radial coordinates are obtained by fitting cubic splines along the elliptic mesh lines and interpolating at a number of axial stations at which the radius of the corresponding axisymmetric surface is known.



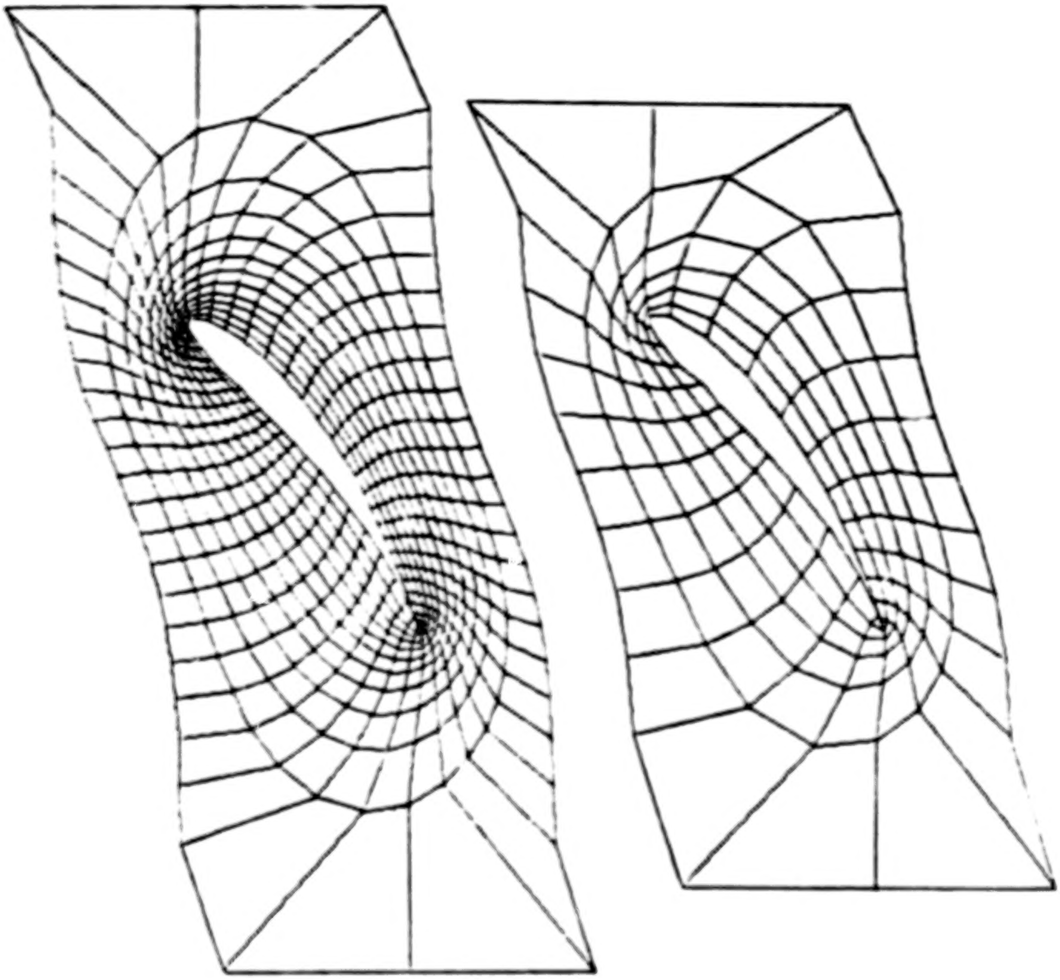
The present method is equally applicable to the blades with blunt (or rounded), wedge and cusp trailing and/or leading edge.


1.4
1.0
0.6
0.2
-0.2
-0.6
-0.7
-0.3
0.1
0.5
0.9
1.3

A disadvantage of the present method is that it is not applicable for the very thick, highly stagered blades which are very closely spaced. This problem can be resolved by using a different form of the mapping function; for example, one which maps a cascade of circles into a cascade of circular arcs instead of a cascade of straight slots.



A sample run shows that it takes 7.3 seconds of CPU time on an IBM 3033 to generate (x, y, z) coordinates of two 3-D grids and to write them on two separate disks. The first (coarse) grid consisted of 27x9x9 points and the second (refined) grid has 51x15x17 points.



REFERENCES

1. Dulikravich, D. S.: Numerical Calculation of Inviscid Transonic Flow Through Rotors and Fans. Ph.D. Thesis, Cornell Univ., 1979.

2. Spiegel, M. R.: Mathematical Handbook of Formulas and Tables. Schaum's Outline Series, McGraw-Hill Book Co., 1968, p. 127.

THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS

Lars-Erik Eriksson
The Aeronautical Research Institute of Sweden (FFA)

ABSTRACT

In this work, a direct algebraic method has been developed and applied to generate three-dimensional grids around wing-body configurations. The method used is a generalized transfinite interpolation method which generates the desired coordinate transformation using geometric data only on the boundaries of the domain of interest. The geometric data that can be specified includes not only coordinates on the boundaries but also out-of-surface parametric derivatives that give a very precise control over the transformation in the vicinity of the surface. In addition to this, the method gives good control over the stretching of the mesh between different boundaries.

The topology of the transformation chosen for the wing-body problem is of a novel type which gives a grid that wraps around not only the leading edge of the wing, but also the wing tip. The body is represented by a deformation of the plane-of-symmetry.

For mesh verification, a simple finite element type algorithm is used to solve the Laplace equation (incompressible flow) on the mesh in question. By varying the details of the matrix evaluation process it is possible to obtain solutions which are more or less dependent on the global mesh properties and thereby get a measure of the "quality" of the mesh. This is essential for applications where for example finite volume methods are used, since these methods depend on smooth global properties of the mesh.

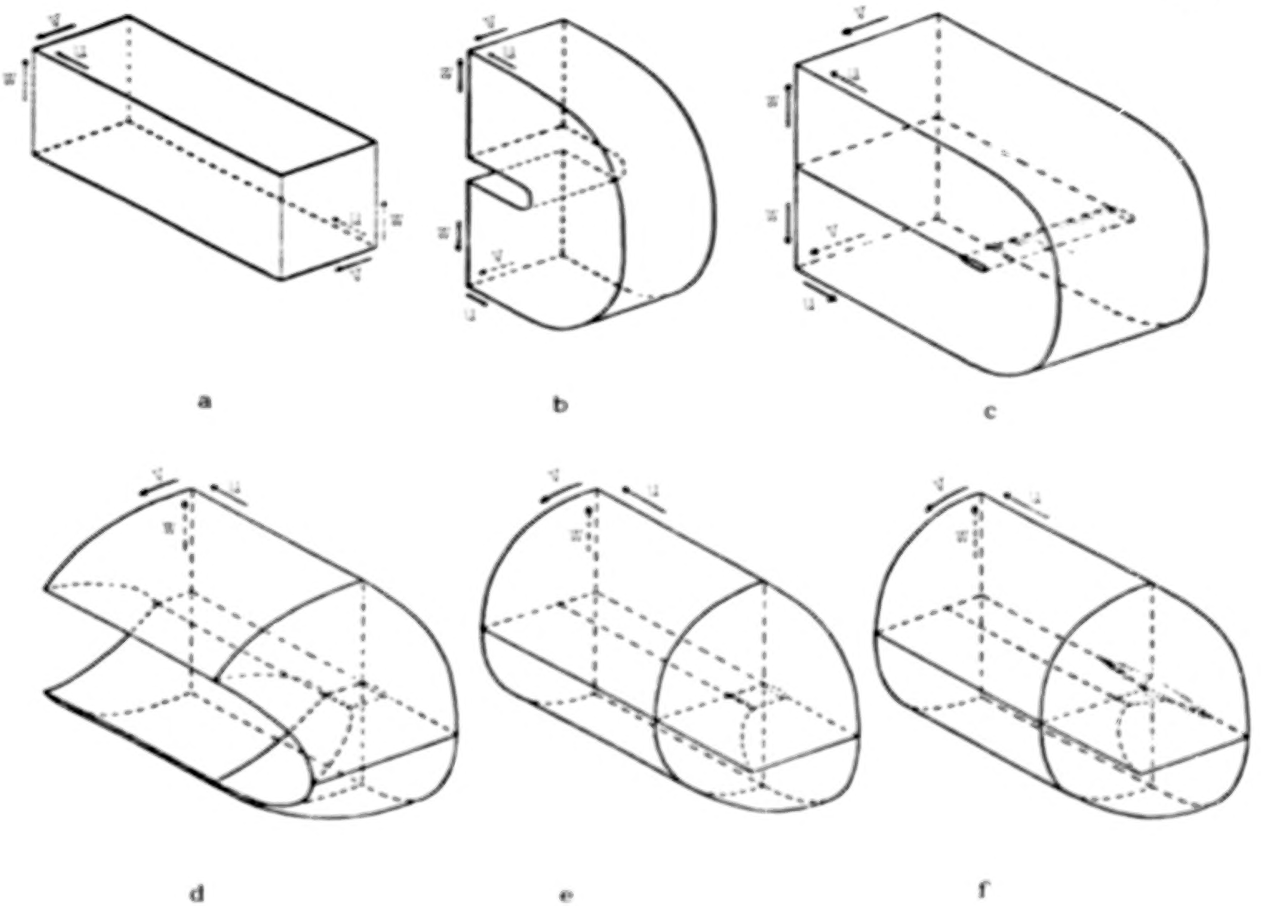


Fig. 1. Topology of wing-body transformation.

The topology of the transformation used for the wing-body problem is outlined by these schematic figures. In the top left figure, the computational or parametric domain is shown with u,v,w as arbitrary parameters. The top middle and right figures illustrate the first step in the transformation, which is a "folding" process to obtain a wing-like inner boundary and an internal branch cut behind the wing trailing edge. The next step is shown in the bottom left and middle figures and consists of another "folding" process in the spanwise direction of the wing. This results in the collapse of a surface into another internal branch cut outside the wing and wake. The last figure shows the third and last step, introducing the body. This is done by deforming the plane-of-symmetry boundary and displacing the wing and wake appropriately.

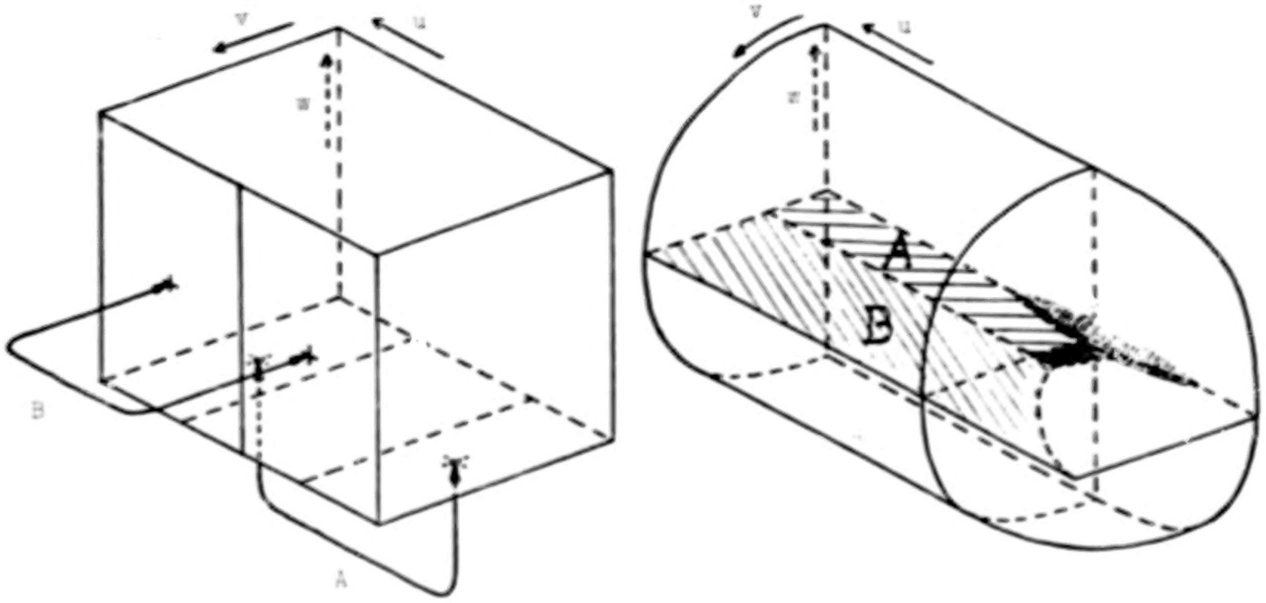
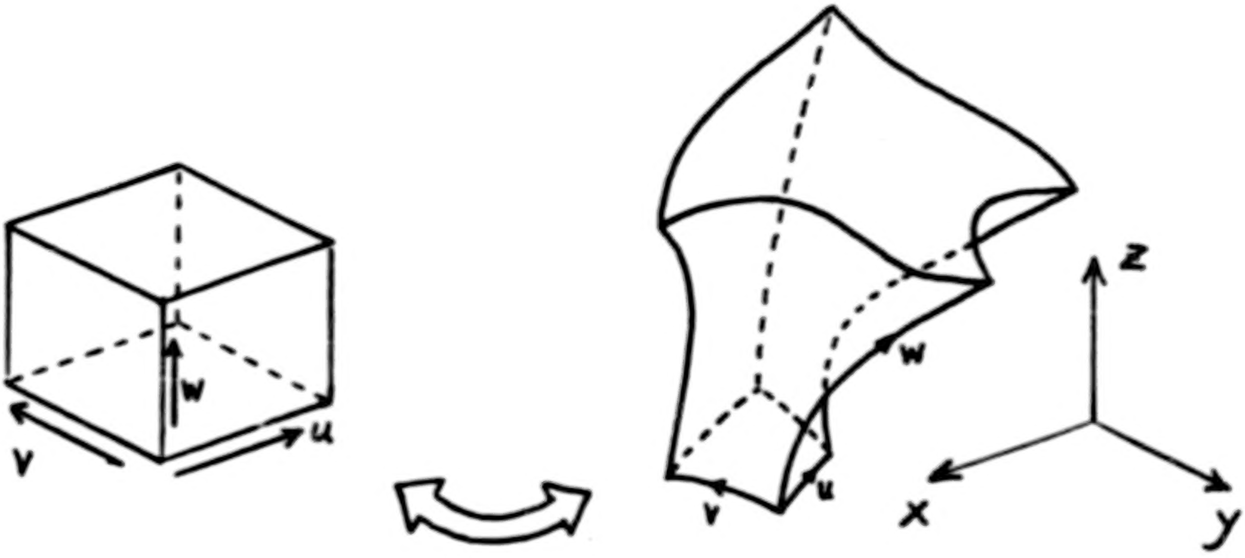


Fig. 2. Internal branch cuts.

The internal branch cuts are shown in both the parametric domain and the physical domain. For actual flow computations in the parametric domain, the usual boundary conditions have to be supplemented by appropriate continuity conditions at the branch cuts.



$$\begin{cases} x = x(u,v,w) \\ y = y(u,v,w) \\ z = z(u,v,w) \end{cases}$$

Fig. 3. Transfinite Interpolation

To generate the desired transformation, a generalized transfinite interpolation method is used. This procedure (which alternately can be viewed as a generalized spline interpolation procedure) gives a transformation by interpolating geometric data from the six boundaries of the parametric domain into the interior of this domain. The geometric data needed for this method consists of coordinates and out-of-surface parametric derivatives. With appropriate choices of coordinates and derivatives it is also possible to generate some boundaries automatically, thus reducing the number of boundaries that have to be specified.

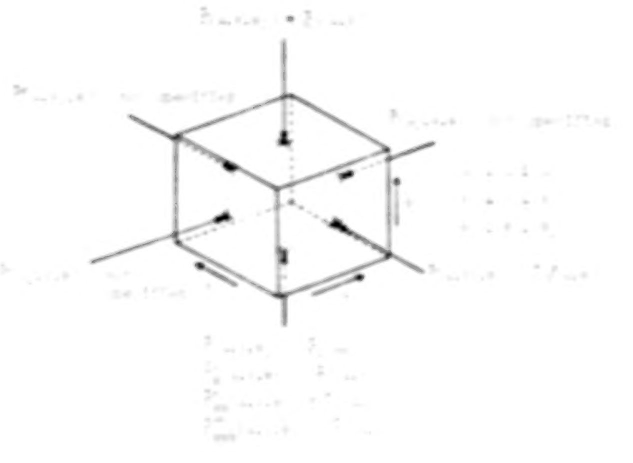


Fig. 4. Application of the transfinite interpolation method to the wing-body transformation.

This figure shows the parametric domain and the geometric data needed to generate the wing-body transformation. The vector valued function $\bar{f}(u,v,w)$ is here the desired transformation $[x(u,v,w),y(u,v,w),z(u,v,w)]$ and is specified only on three of the six boundaries: the plane $w=w_1$ (=the transformed wing and wake surface), the plane $w=w_2$ (=the transformed outer boundary) and the plane $v=v_1$ (=the transformed body and plane-of-symmetry boundary). On the plane $w=w_1$, there are also out-of-surface parametric derivatives $\bar{f}'_w$, $\bar{f}''_{ww}$, $\bar{f}'''_{www}$ specified. These parametric derivatives are essential to give a precise control over the transformation near the wing surface and are also necessary to generate automatically the geometric data for the remaining three boundaries. The choice of first, second and third derivatives is arbitrary (it is possible to specify any number of derivatives), but has been found to give good results.

To generate the transformation $\bar{F}(u,v,w)$ it is necessary to introduce blending functions $\beta_1(v)$: $v_1 \leq v \leq v_2$ and $\gamma_1(w)$, $\gamma_2(w)$, $\gamma_3(w)$, $\gamma_4(w)$, $\gamma_5(w)$: $w_1 \leq w \leq w_2$ with conditions:

$$
\begin{array}{llllll}
\beta_1(v_1) = 1 & \gamma_1(w_1) = 1 & \gamma_2(w_1) = 0 & \gamma_3(w_1) = 0 & \gamma_4(w_1) = 0 & \gamma_5(w_1) = 0 \\
\beta_1(v_2) = 0 & \gamma_1'(w_1) = 0 & \gamma_2'(w_1) = 1 & \gamma_3'(w_1) = 0 & \gamma_4'(w_1) = 0 & \gamma_5'(w_1) = 0 \\
 & \gamma_1''(w_1) = 0 & \gamma_2''(w_1) = 0 & \gamma_3''(w_1) = 1 & \gamma_4''(w_1) = 0 & \gamma_5''(w_1) = 0 \\
 & \gamma_1'''(w_1) = 0 & \gamma_2'''(w_1) = 0 & \gamma_3'''(w_1) = 0 & \gamma_4'''(w_1) = 1 & \gamma_5'''(w_1) = 0 \\
 & \gamma_1(w_2) = 0 & \gamma_2(w_2) = 0 & \gamma_3(w_2) = 0 & \gamma_4(w_2) = 0 & \gamma_5(w_2) = 1
\end{array}
$$

The transfinite interpolation scheme is then defined by

1. $\bar{F}^*(u,v,w) = \gamma_1(w)\bar{F}_1(u,v) + \gamma_2(w)\bar{F}_1(u,v) + \gamma_3(w)\bar{F}_1(u,v) + \gamma_4(w)\bar{F}_1(u,v) + \gamma_5(w)\bar{F}_2(u,v)$

2. $\bar{F}(u,v,w) = \bar{F}^*(u,v,w) + \beta_1(v)[\bar{F}_3(u,w) - \bar{F}^*(u,v_1,w)]$

The choice of blending functions has to be made with care, since they have a direct influence on the "stretching" of the transformation between different boundaries.

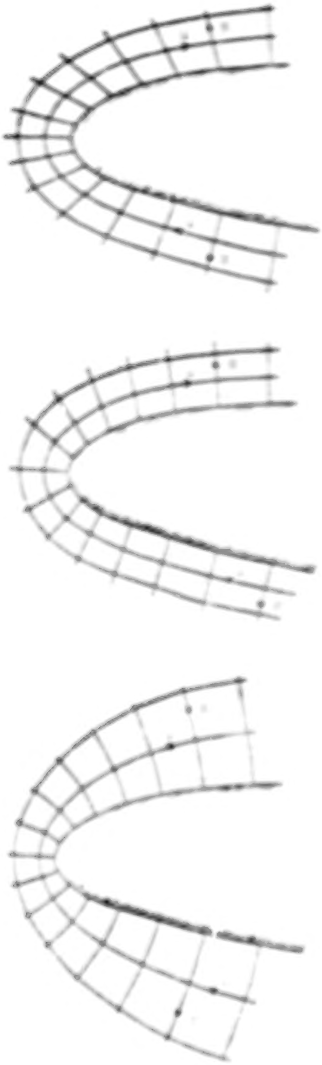


Fig. 5. Effect of out-of-surface parametric derivatives.

These three figures show the importance of the out-of-surface parametric derivatives that must be specified on the wing and wake boundary. The top figure illustrates the case where only the first derivative $[x'_w, y'_w, z'_w]$ is specified. This derivative determines the direction of the outgoing grid lines (lines with constant u and v) and the spacing between successive grid surfaces (surfaces with constant w). To obtain a univalent transformation, the derivative must be adapted to the surface geometry and also vary smoothly. An obvious way to adapt the derivative is to make it orthogonal to the surface (middle figure). If the radius of curvature varies very rapidly however, this simple solution does not work very well, because it requires an excessive concentration of grid points in the critical region. A better solution is to specify higher derivatives, for example $[x''_{ww}, y''_{ww}, z''_{ww}]$ and $[x'''_{www}, y'''_{www}, z'''_{www}]$. With these derivatives, it is possible to obtain a great variety of transformations near the wing surface. The bottom figure shows an example where the transformation is approximately conformal in the vicinity of the wing.

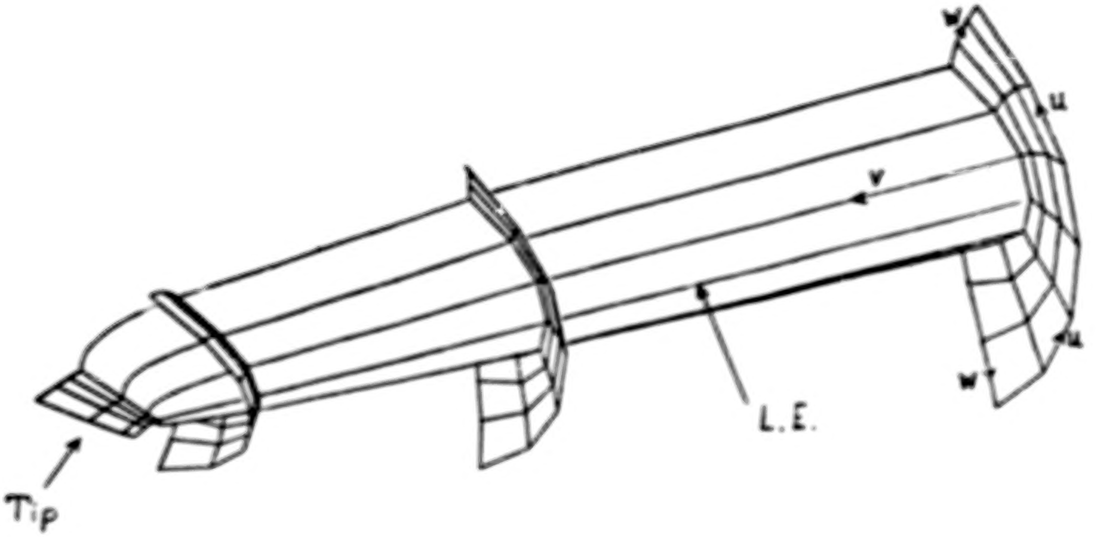


Fig. 6. Detail of typical grid in the leading edge/tip region.

In order to obtain the desired behaviour of the transformation in the leading edge/tip region of the wing. It is necessary to specify the out-of-surface parametric derivatives as smooth functions of both the in-surface parameters u and v. This figure is an oblique view of a typical grid in this region and shows how the constant-v surfaces gradually collapse into the branch cut outside the wing tip as the v-parameter is increased towards the maximum value.

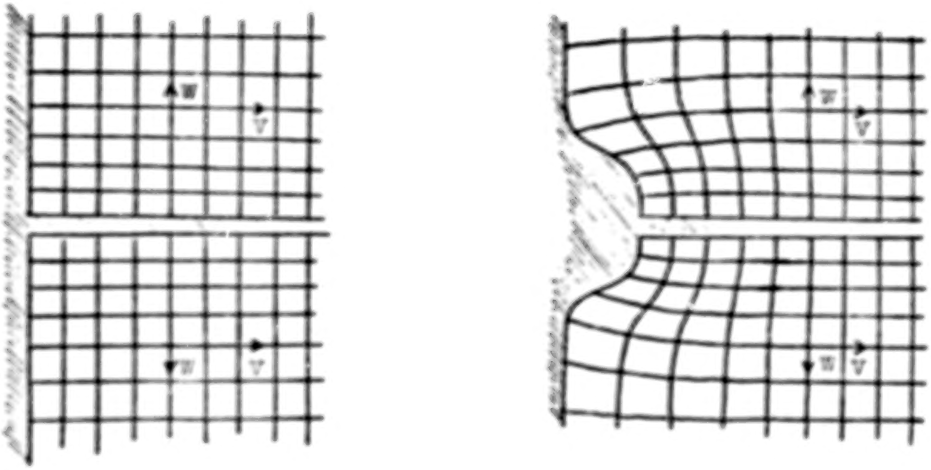


Fig. 7. Deforming the plane-of-symmetry to represent the body.

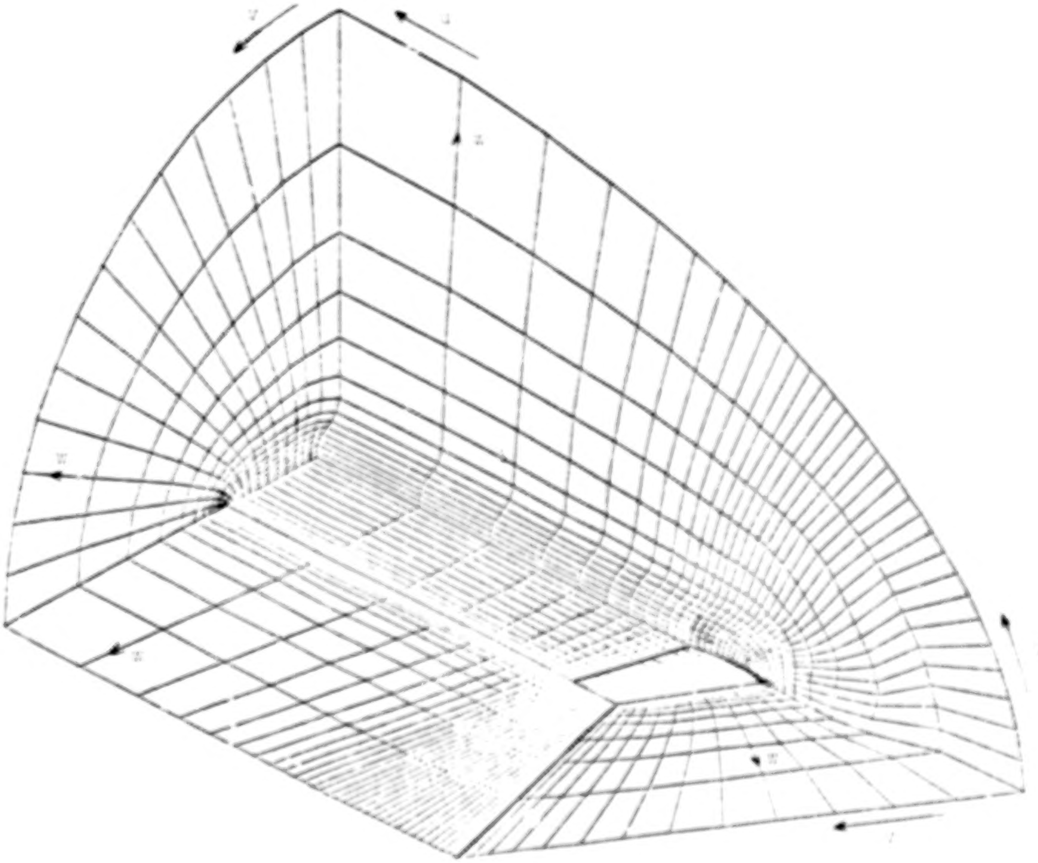
This figure shows the effect on the transformation of deforming the plane-of-symmetry to simulate a half-body. The wing and wake surface is translated outwards and the deformation is interpolated into the domain in a smooth manner that is determined by the blending function $\beta_1(v)$. Since the deformed plane-of-symmetry is specified in terms of coordinates, it is also possible to concentrate the grid lines around the body as shown by the figure.

Shown in figure 8 are several views of a wing-body grid. Figures 8(a) and 8(b) show from two perspective views:

- the upper half of the wing and wake surface
- the upper half of the deformed plane-of-symmetry
- the internal branch cut outside the wing and wake
- one of the downstream boundaries
- the constant-u surface that emanates from the wing leading edge

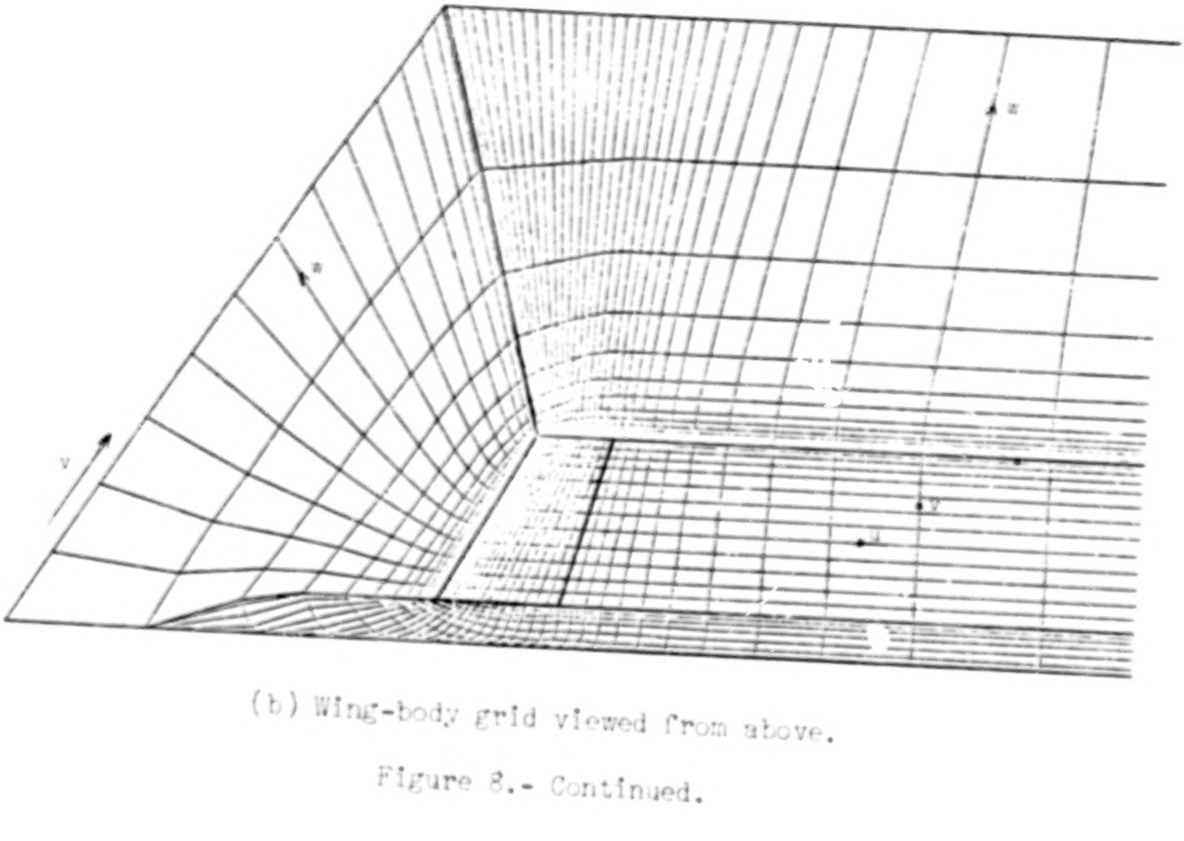
Figures 8(c) and 8(d) show a constant-v surface of the same grid as in figure 8(a). This surface emanates approximately from the mid section of the wing.

Figures 8(e) and 8(f) show two constant-u surfaces of the same grid as in figure 8(a). These surfaces emanate from the upper and lower x/c = 0.25 lines.



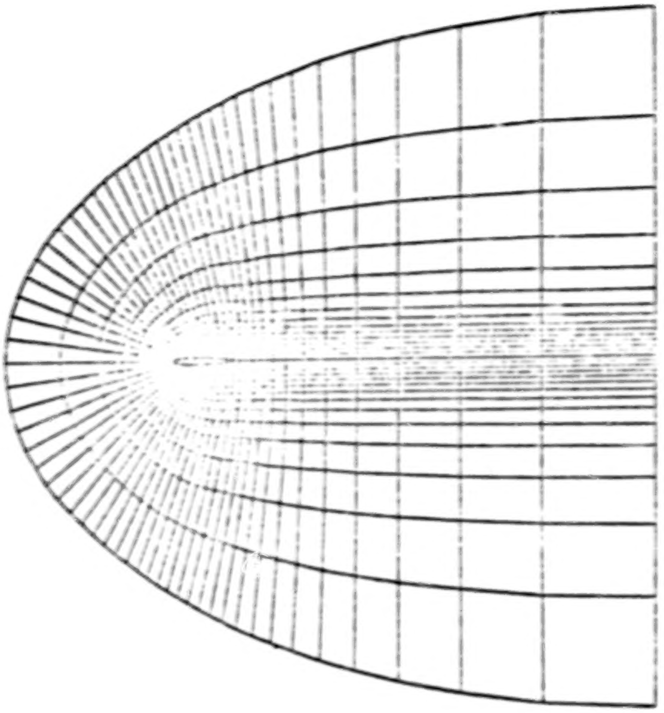
(a) Wing-body grid viewed from below.

Figure 8.- Wing-body grid.

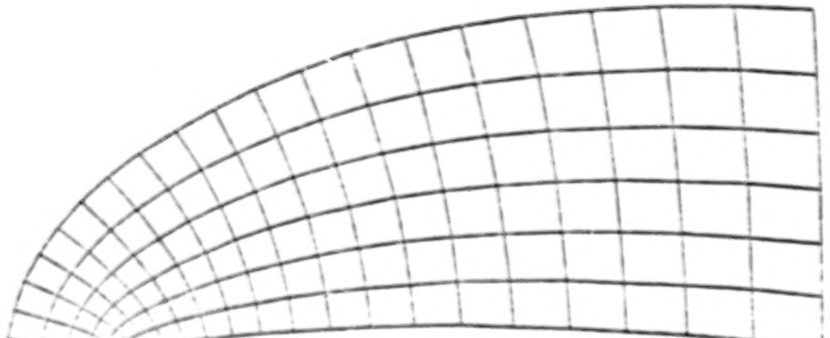


(b) Wing-body grid viewed from above.

Figure 8.- Continued.

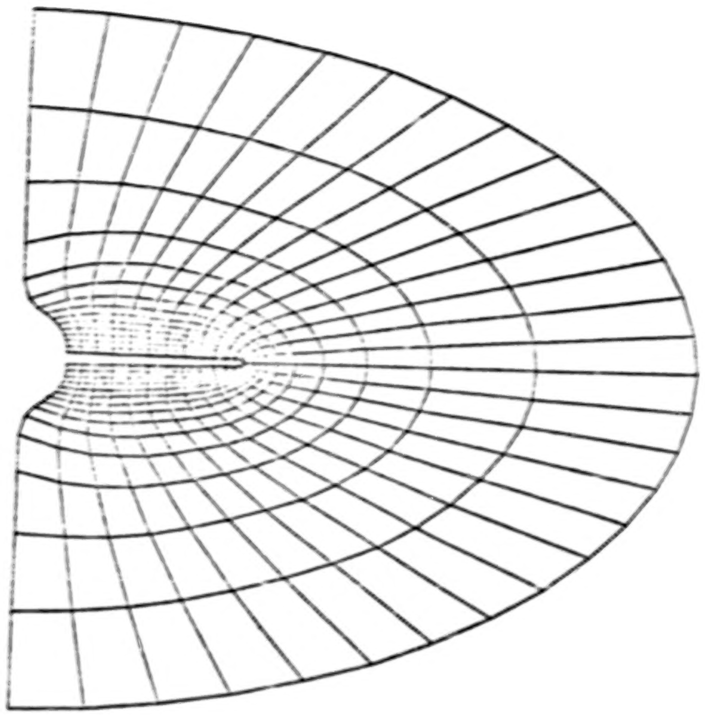


(c) Planar view of wing-body grid in wing wake region.

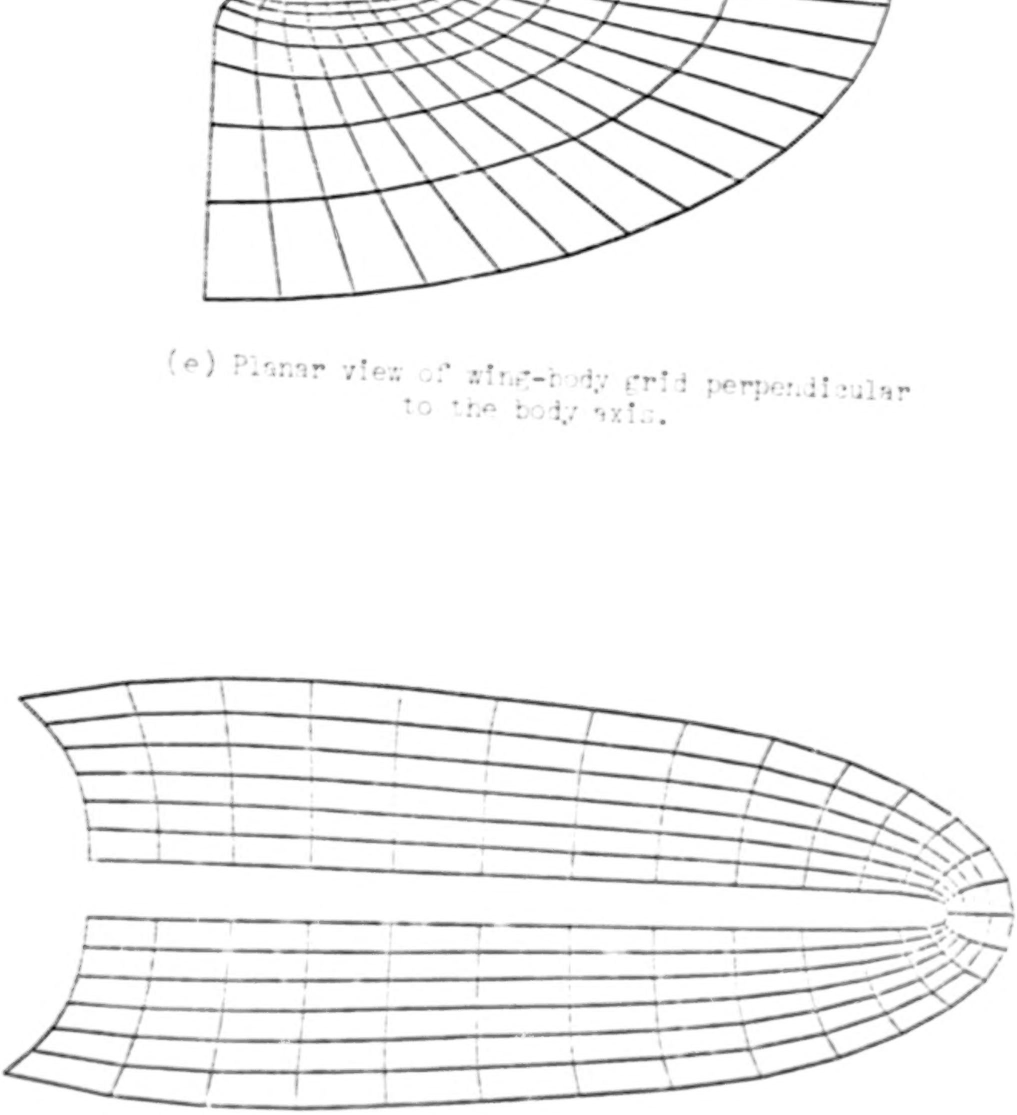


(d) Enlarged view of grid at wing surface.

Figure 8.- Continued.



(e) Planar view of wing-body grid perpendicular to the body axis.



(f) Enlargement of planar view.

Figure 8.- Concluded.

AN INVESTIGATION INTO GRID PATCHING TECHNIQUES*

C. R. Forsey, M. G. Edwards, and M. P. Carr

Aircraft Research Association
Manton Lane
Bedford, England

ABSTRACT

In the past decade significant advances have been made using flow field methods in the calculation of external transonic flows over aerodynamic configurations. It is now possible to calculate inviscid transonic flow over three-dimensional configurations by solving the potential equation. However, with the exception of the Transonic Small Disturbance methods which have the advantage of a simple cartesian grid, the configurations over which it is possible to calculate such flows are relatively simple (eg wing plus fuselage). The major reason for this is the difficulty of producing compatibility between grid generation and flow equation solutions. The main programs in use, eg Jameson in US and Forsey in UK, use essentially analytic transformations for prescribed configurations and, as such, are not easy to extend. Whilst there is work in progress to extend this type of system to a limited extent, our longer term effort is directed towards a more general approach. This approach should not be restricted to producing grid systems in isolation but rather a consideration of the overall problem of flow field solution.

This paper describes one approach to this problem.

---

*This work has been carried out with the support of Procurement Executive, Ministry of Defence.

GRID GENERATION

GENERAL APPROACH

1. Grid generation, or equivalent, vital to solution of general flow field problems.

2. It is not obvious which technique to use.

3. Various methods being explored

a) Non-aligned grid	Catherall	R.A.E
b) Aligned grid with global solution for grid with control function	Roberts	British Aerospace
c) Aligned grid with local grids patched together	Forsey	A.R.A

FIGURE 1 GENERAL APPROACH TO GRID GENERATION

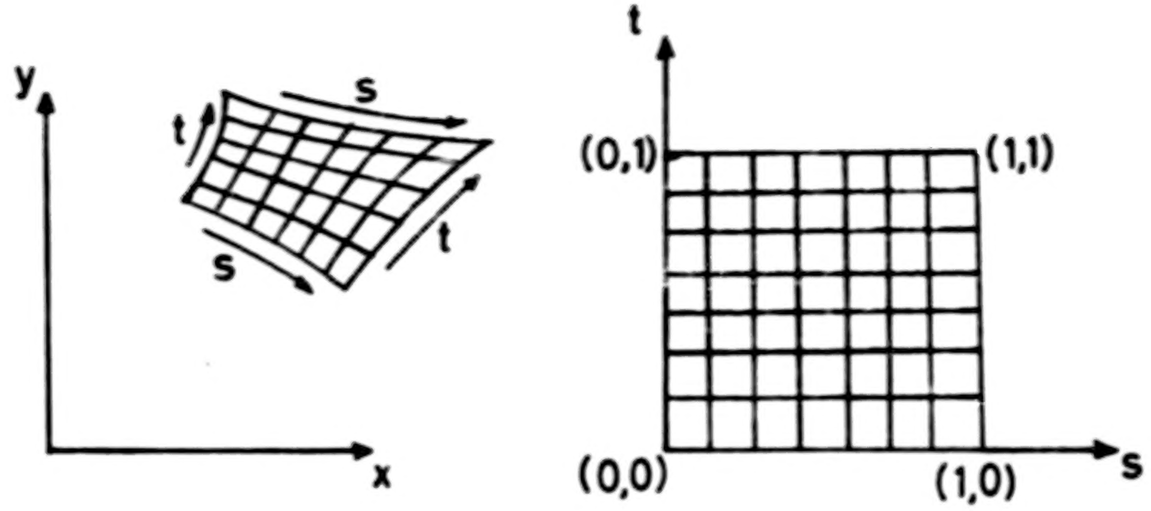
1. GENERAL APPROACH TO GRID GENERATION

Accepting the requirement to solve a set of discretised partial differential equations on the nodes of a suitable grid, then some method of grid generation is vital. Since there are a number of ways in which one can tackle this problem and, at the moment, no one technique appears sufficiently superior to others, it would appear judicious to attempt more than one approach. Therefore, a program of work is being undertaken in various UK establishments to investigate suitable techniques and the method described in this paper is part of this overall project.

Probably the first question one poses when considering the requirement of a computing grid is whether or not to align the grid with the surface. Catherall at RAE is investigating the non-aligned grid concept. The grid, being cartesian, can be generated in a straight forward manner with the major problems being the complicated application of boundary conditions and the general 'housekeeping' for complex configurations. However, extra components can be added fairly easily and it should be versatile.

If an aligned grid is considered mandatory then the application of boundary conditions becomes much simpler and grid generation becomes a major problem. Roberts at British Aerospace is attempting to produce a method of grid generation for general three-dimensional configurations by producing a global solution of a set of partial differential equations. The introduction of mapping singularities is used to control the distribution of grid points using discretisation based on triquintic splines.

The work described here investigates a method some way between these two techniques. The requirement for an aligned grid system is accepted but with the flow field divided into segments, each segment with its own rather straight forward grid system. The surface boundary conditions are easy to apply but the main problem is one of solving the flow equations through the boundaries where the segments are patched together. This approach will now be described in more detail.



$$f(s,t)=(1-s)f(0,t)+sf(1,t)+(1-t)f(s,0)+tf(s,1)$$
$$-(1-s)(1-t)f(0,0)-(1-s)tf(0,1)-s(1-t)f(1,0)-stf(1,1)$$

where $f(s,t)=x(s,t)$ or $y(s,t)$

and $f(s,0)$, $f(s,1)$, $f(0,t)$, $f(1,t)$

define the four sides as parametric functions of s and t.

FIGURE 2 BASIC ISOPARAMETRIC MAPPING

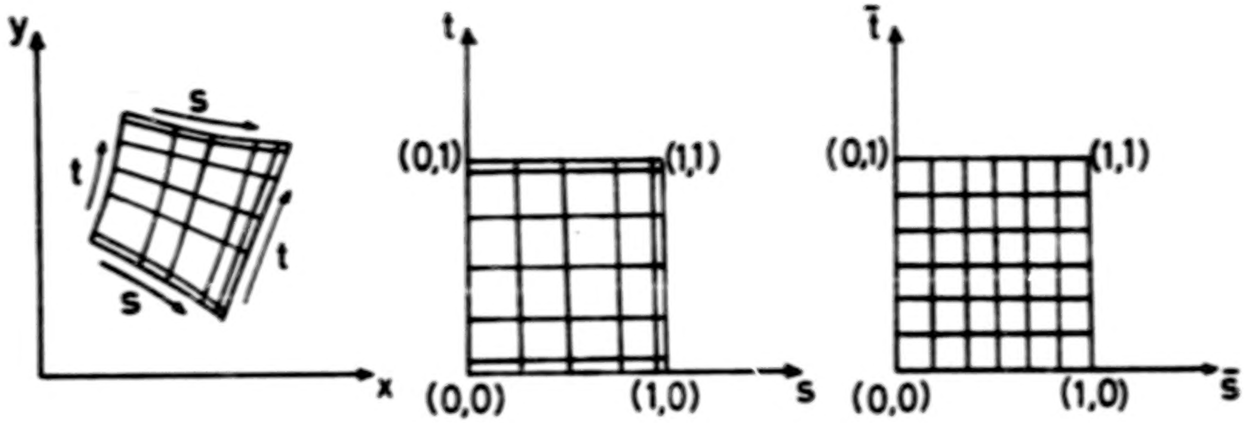
2. BASIC ISOPARAMETRIC MAPPING

The patching technique consists of splitting up regions of interest into a series of quadrilateral segments which are patched together across common boundaries. For the grids in each segment any convenient method of grid generation may be used. However, as it is necessary to match the grids across common boundaries and to maintain some control over the grid spacing in these regions, which is most conveniently done using interactive graphics, a grid generation technique combining simplicity with minimal computer requirements is needed.

One such method, which has been used extensively in finite element work, is the isoparametric or blending function method. This method consists of defining the x and y coordinates of points within a quadrilateral as parametric functions (f) of two parameters (s,t) where s = 0,1 and t = 0,1 define the sides of the quadrilateral in parametric space. If the values of f are defined along s = 0,1 and t = 0,1 (ie the point distributions along the sides are prescribed) then the blending function f defines internal points as a smooth blending between these boundary values. Taking equal intervals in s and t then defines the grid lines within the quadrilateral.

The values of f along s = 0,1 and t = 0,1 are defined by cubic spline curve fits of f vs.s or t where s and t are taken as the arc lengths along the appropriate sides.

The blending function used in the present patching method is the lowest order blending function which is a bilinear blending. However, higher order blendings (eg cubics) could be used with very little increase in computing time and the extra degrees of freedom then used to define the shape of some of the internal grid lines or the slope of the grid lines at the boundaries.

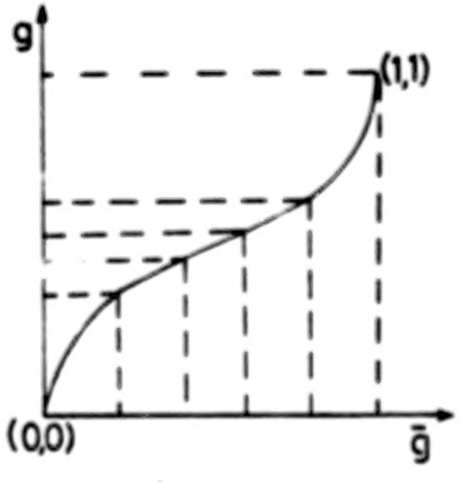

y
x
s
t
t
s
t
(0,1)
(1,1)
(0,0)
(1,0)
s
$\bar{t}$
(0,1)
(1,1)
(0,0)
(1,0)
$\bar{s}$

$$g=A\bar{g}^3+B\bar{g}^2+C\bar{g}+D$$

where $g(\bar{g})=s(\bar{s})$ or $t(\bar{t})$

and $g=0$ at $\bar{g}=0$

$g=1$ at $\bar{g}=1$



$dg/d\bar{g}$ at $\bar{g}=0,1$

specified by user

FIGURE 3 STRETCHED ISOPARAMETRIC MAPPING

3. STRETCHED ISOPARAMETRIC MAPPING

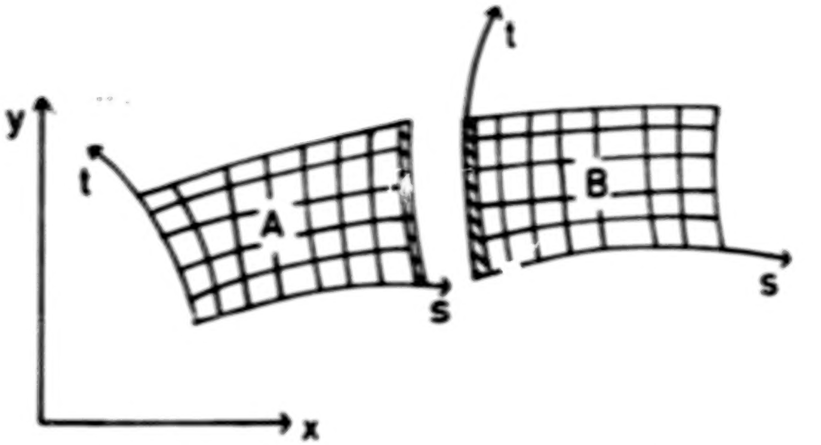
The basic isoparametric mapping produces grids which vary smoothly between opposing sides of the quadrilateral. However, for equally spaced intervals in parametric (s,t) space the corresponding grid lines in physical (x,y) space, although curvilinear, are still equally spaced. It is convenient to have the facility for packing grid lines near specific quadrilateral boundaries or near the middle of the quadrilaterals.

Hence, preliminary stretching transformations are applied to the s and t coordinates. One simple stretching which gives considerable user control is to make s and t cubic functions of some other parameters $\bar{s}$ and $\bar{t}$. Taking equal intervals in $\bar{s}$ and $\bar{t}$ then results in unequally spaced intervals in s and t. By appropriate choice of the derivatives $ds/d\bar{s}$ and $dt/d\bar{t}$ at each end of the cubic it is possible to pack points towards either end (one value of $ds/d\bar{s} < 1$, the other value > 1), towards the middle (both values of $ds/d\bar{s} > 1$), or towards both ends (both values of $ds/d\bar{s} < 1$).

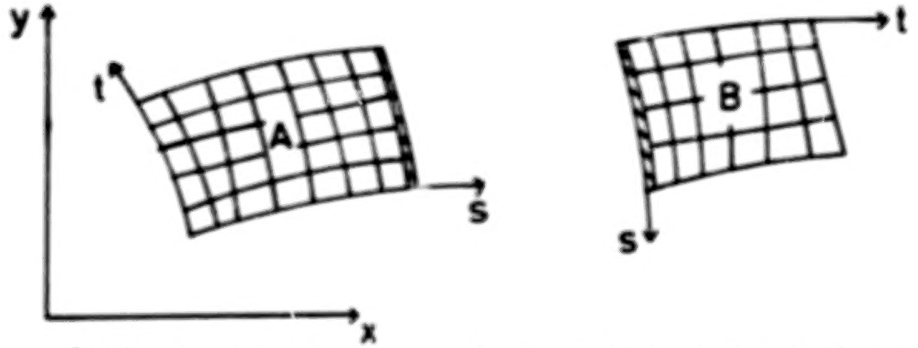
For all cases except the last a single cubic appears adequate. In the last case, however, attempting to pack points towards both ends usually results in a grid which has very fine spacing near both ends but which then suddenly jumps to much wider spacing near the middle. This seems to be due to an inability to control the slope of the cubic near the middle where the slope remains much the same regardless of the slopes imposed at each end. One solution which we are currently using for this case is to replace the single cubic by a cubic spline curve through 4 points. The slopes are still specified at the end pair of points and the middle pair of points are chosen to control the slope of the curve near the middle.

In practice, the stretching parameters (ie $ds/d\bar{s}$ and $dt/d\bar{t}$ at each end plus the two middle points for the cubic spline stretching) are chosen interactively by the user with the aid of interactive graphics.

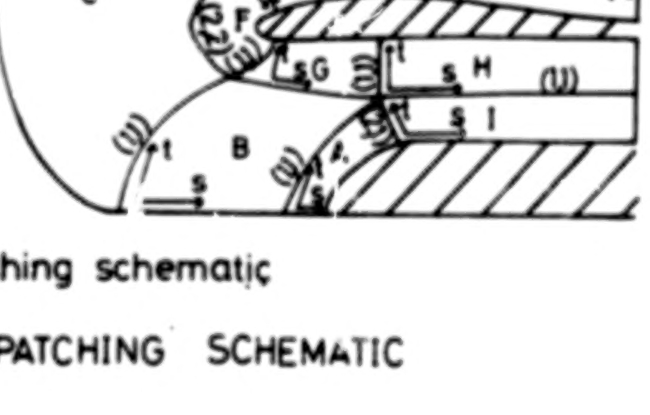
In order to increase flexibility still further different values of the stretching parameters can be specified on opposing sides of the quadrilateral and a linear variation between these values is used for all internal grid lines between these two sides.



(1,1)patch: s=1 segment A patched to s=0 segment B



(1,2)patch: s=1 segment A patched to t=0 segment B


s
t
(1,1)
D
s
t
(2,2)
E
s
t
C
s
F
(2,2)
s G
t
s H
(1,1)
s I
(1,1)
t
B
s
A
s

Intake patching schematic

FIGURE 4 PATCHING SCHEMATIC

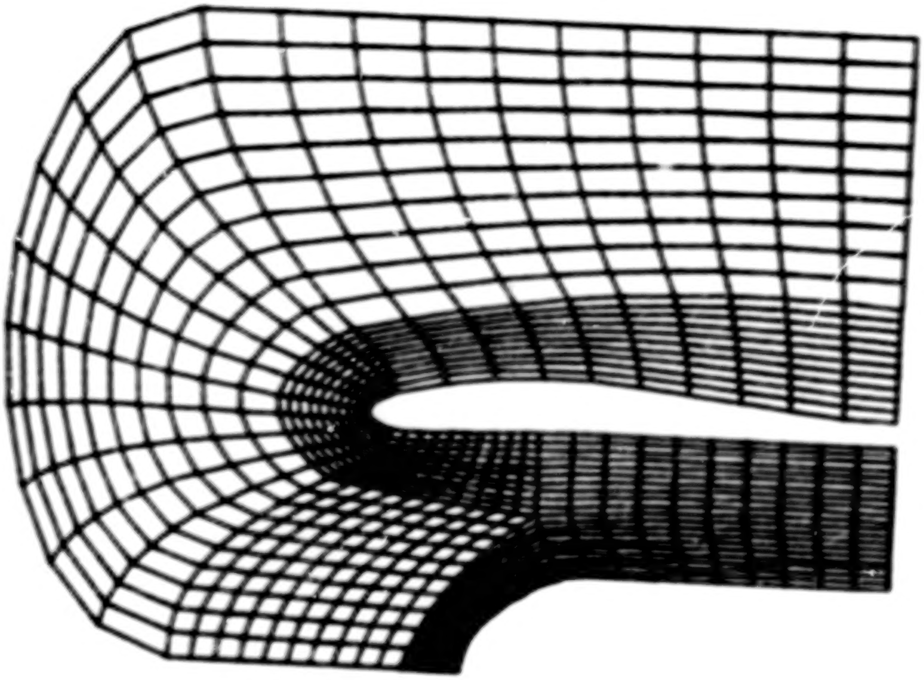
PATCHING SCHEMATIC

Individual segments, each with their own local isoparametric grids are joined (or 'patched') along common boundaries until the whole region of interest has been covered. Grid lines in two adjoining segments must meet on their common boundary which implies that the same number of points and the same stretching functions are used on both sides of this boundary. However, the grid lines may change direction through this boundary, i.e. they need not be smooth. Instead, special boundary conditions are applied on patched boundaries to ensure flow continuity. Sides of segments corresponding to solid surfaces or free stream conditions also have appropriate boundary conditions applied.

Since maximum flexibility is required when choosing the way the region of interest is split up into segments, it is necessary to allow any side of one segment to patch to any side of an adjoining segment. Two typical patches, designated (1,1) patches and (1,2) patches are illustrated and there are several others. In principle, different types of patch should not significantly increase the difficulty of applying the appropriate patch boundary conditions. However, in practice they considerably increase the general program housekeeping needed and in the current program not all types of patches have been allowed for as yet.

At present the way the region is divided into segments is controlled by user input although eventually it is hoped to (at least partially) automate this process. Initially, all solid surfaces (eg aerofoil surfaces etc) are defined accurately and then the user defines the segment boundaries, some of which are parts of solid surfaces and some separate hand drawn curves. A schematic showing a typical setting up procedure for an inlet with central bullet is shown. The orientations of each segment and the type of each patch are indicated on the schematic.

Basic unstretched grid



Final stretched grid

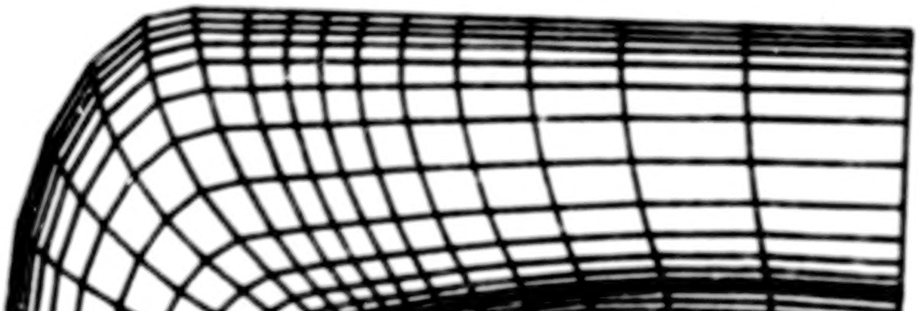


FIGURE 5 EXAMPLE OF GRID-INTAKE

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL
TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS
FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED
BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B1
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR
FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR
FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO
REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY
COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX
FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN
SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING
UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

5. EXAMPLE OF GRID - INTAKE

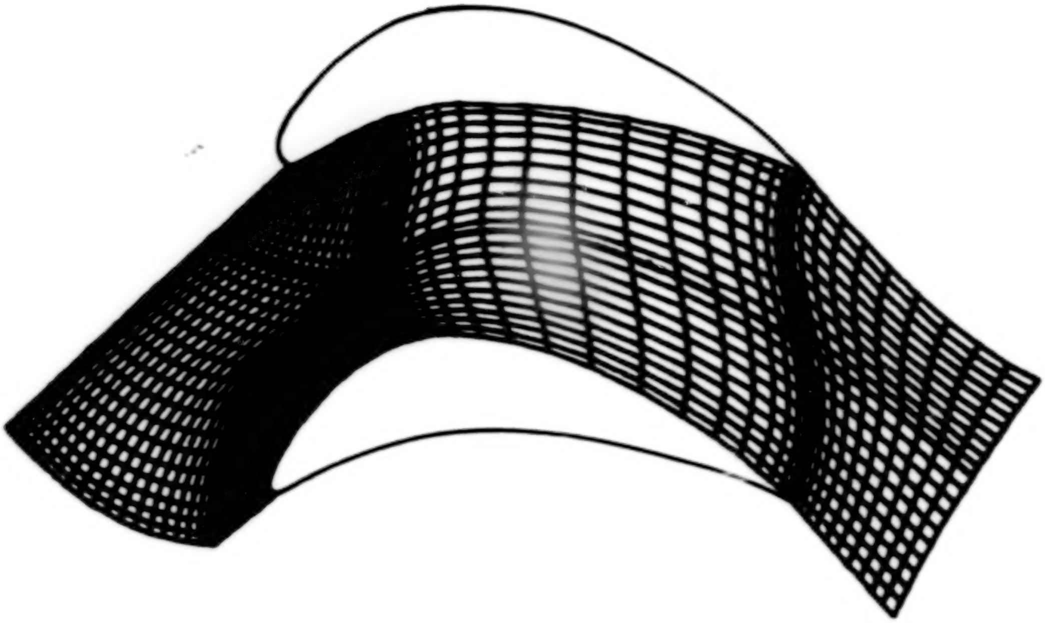
When the region has been divided into segments to the user's satisfaction, the next step is to define the number of points in the s and t directions and the corresponding unstretched grids in each segment. This gives a general idea of what the final overall grid will be like. An example is given at the top of Fig 5 for the intake with central bullet.

Various stretchings are then applied to the s and t coordinates in each segment to remove sudden changes in the width of grid intervals (particularly across patched boundaries) and to pack grid lines in regions where the flow is expected to vary rapidly. The stretching parameters are modified, and the resulting grids displayed, interactively using a graphics terminal. The final overall grid is shown at the bottom of Fig 5. Approximately two days work was required to produce this grid from scratch and only a small amount of computer time was required on a modest Prime 400 computer linked to a Tektronix 4051 graphics terminal.

A few comments regarding the choice of segments for this example may be useful. Because of the nature of the blending functions used, the easiest way to ensure that grid lines are approximately normal to solid surfaces is to choose the shape of the patch boundaries which join such surfaces to be nearly normal to the surfaces concerned as has been done in segments A and F. Furthermore, in order to accurately model the cowl surface boundary condition and the channel flow between the cowl and the bullet, a fine inner grid is patched to a sparse outer grid but stretchings are used to ensure that a sudden change in grid spacing does not occur at the patch boundaries.

It will be noticed that at one point five patch boundaries (ie grid lines) meet rather than the usual four. We feel that this should give no particular problems especially if the point is in a region where there are no flow singularities. (There is some evidence that putting such a point at a stagnation point of the flow can lead to difficulties).

grid type 1



grid type 2

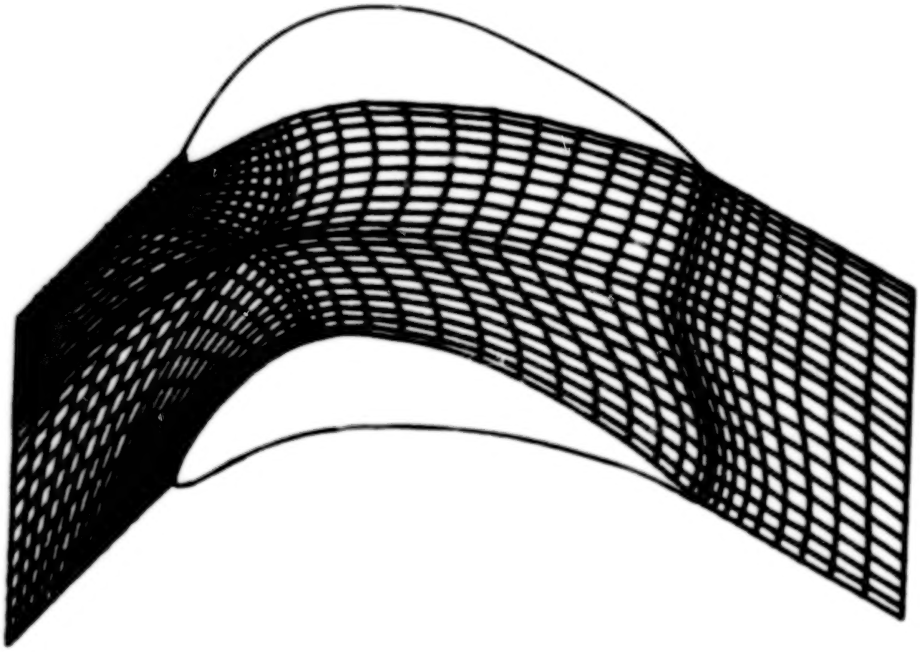


FIGURE 6 EXAMPLE OF GRID - CASCADE

6. EXAMPLE OF GRID - CASCADE

The region of interest may be divided into segments, under user control, in any number of ways and Fig 6 illustrates two different grids for a typical cascade flow problem. The generating segments are marked on each grid. In the two cases different numbers of segments are used with individual segments having quite different shapes. Although there is no restriction, in principle, on the number of segments which may be used, there is some evidence that, at the current stage of development, the convergence rate for the solution of the flow equation decreases with increasing numbers of segments.

The top grid shows the first attempt at a cascade grid. In this attempt the main criterion in choosing the segments was to produce a fine grid spacing around the leading edges of the two aerofoils, a region where many previous cascade grids have been deficient. Again, notice that at one point five patch boundaries (ie grid lines) meet rather than the more usual four.

However, after producing this grid it was realised that it would be impossible to use periodic boundary conditions across the two lines upstream of the leading edges of the two aerofoils without some form of interpolation. This was because the upper and lower lines are formed by different combinations of segment boundaries and hence have different numbers of points and different point spacings. The same argument applies to the two lines downstream of the trailing edge of the two aerofoils. Since neither pair of lines is intended to represent actual streamlines, periodic boundary conditions are the only correct boundary conditions which may be applied across these lines. Hence, the lower grid was produced to try and overcome this restriction without significantly compromising the other advantages of the first grid.

This perhaps illustrates the interactions between grid generation methods and flow calculation methods as the two cannot really be studied independently.

GENERAL TENSOR FORM OF THE FULL POTENTIAL EQUATION

$$\underline{(c^2-q^2)\,v^i v^j \frac{\partial^2 \phi}{\partial r^i \partial r^j}} + c^2 (q^2 g^{ij} - v^i v^j) \frac{\partial^2 \phi}{\partial r^i \partial r^j} + \frac{c^2 q^2}{\sqrt{g}} \frac{\partial \phi}{\partial r^j} \frac{\partial}{\partial r^i} (\sqrt{g} g^{ij})$$

$$-\frac{q^2}{2} v^i v_j v_k \frac{\partial g^{jk}}{\partial r^i} - q^2 v^i v^j \frac{\partial^2 x^1}{\partial r^i \partial r^j} = 0$$

r^i = coordinates in transform space ($r^1 = r$, $r^2 = s$, $r^3 = t$)
ϕ = perturbation velocity potential ($\Phi = \phi + x$)
$c^2 = \frac{1}{M_\infty^2} + \frac{(\gamma-1)}{2} (1-q^2)$ = square of local speed of sound

g = determinant (g_{ij})
g^{ij} = cofactor (g_{ij}) / determinant (g_{ij})
$g_{ij} = \frac{\partial x^k}{\partial r^i} \cdot \frac{\partial x^k}{\partial r^j}$ = metric tensor

x^k = cartesian coordinates in physical space ($x^1 = x$, $x^2 = y$, $x^3 = z$)
$v^i = g^{ij} v_j$ = contravariant velocity

$v^i = \frac{\partial \Phi}{\partial r^i}$ = covariant velocity

$q^2 = v^i v_i$ = total velocity

BOUNDARY CONDITIONS

1. Free stream boundary condition
   a) v_i = free stream velocity in i direction
   or
   b) $\phi = 0$
2. Solid surface boundary condition
   $v^i = 0$
3. Patch boundary condition

FIGURE 7 FLOW EQUATION

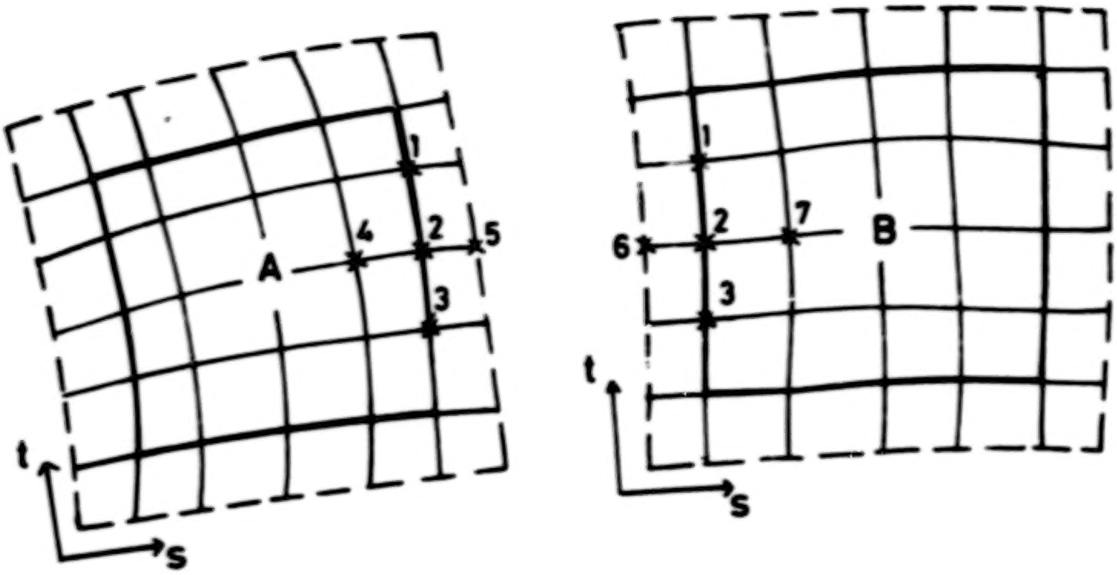
7. FLOW EQUATION

At all interior grid points of each segment the flow is calculated by solving a finite difference approximation to the compressible potential equation. Appropriate boundary conditions, described later, are applied on the four sides of each segment. We find it particularly convenient to work with the general tensor form of the potential equation and its boundary conditions. This is because this form eliminates dependence on the precise nature of the local grid transformations used and because the same equations encompass both two dimensions and three dimensions. Hence, methods developed in tensor form are equally applicable to two and three dimensions.

In Fig 7 the tensor form of the potential equation is shown in terms of a perturbation potential ϕ. It is written in so called rotated form (ie with the principle part split up into streamwise and streamnormal components). The underlined term is the streamwise component of the principle part and it is this term which is backward differenced in supersonic regions. The metric tensor g_{ij} represents a transformation between physical space with coordinates $x^i = w,y,z$ and some arbitrary space with coordinates $r^i = r,s,t$.

This potential equation may be solved by any convenient numerical method. At present we solve it in nonconservative form using a line overrelaxation method. However, it is planned to implement an approximate factorisation scheme in order to improve the convergence rate in the near future.

Three main types of boundary conditions can be applied on the sides of each segment. The first two types: solid surface conditions (ie zero normal flow through the surface) and free stream conditions (ie zero perturbation velocity or zero perturbation potential) are the same as used with non-patched grids. These are applied in a standard way using dummy rows of grid points outside of the relevant boundaries and no further description will be given. The third type: patch boundary conditions are the heart of the patching method and will be described in detail.



(a) 2D flow equation in non-rotated form with

$r^1 = s,\ r^2 = t$

$$q^2(c^2g^{11} - v^1v^1)\phi_{ss} + q^2(c^2g^{22} - v^2v^2)\phi_{tt} = \text{cross derivatives - low order terms}$$

(b) Continuity of normal velocity across common boundary

$$\left[\frac{v}{\sqrt{g^{11}}}\right]_A = \left[\frac{v}{\sqrt{g^{11}}}\right]_B$$

i.e.

$$\left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s + x_s) + \frac{g^{12}}{\sqrt{g^{11}}}(\phi_t + x_t)\right]_A = \left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s + x_s) + \frac{g^{12}}{\sqrt{g^{11}}}(\phi_t + x_t)\right]_B$$

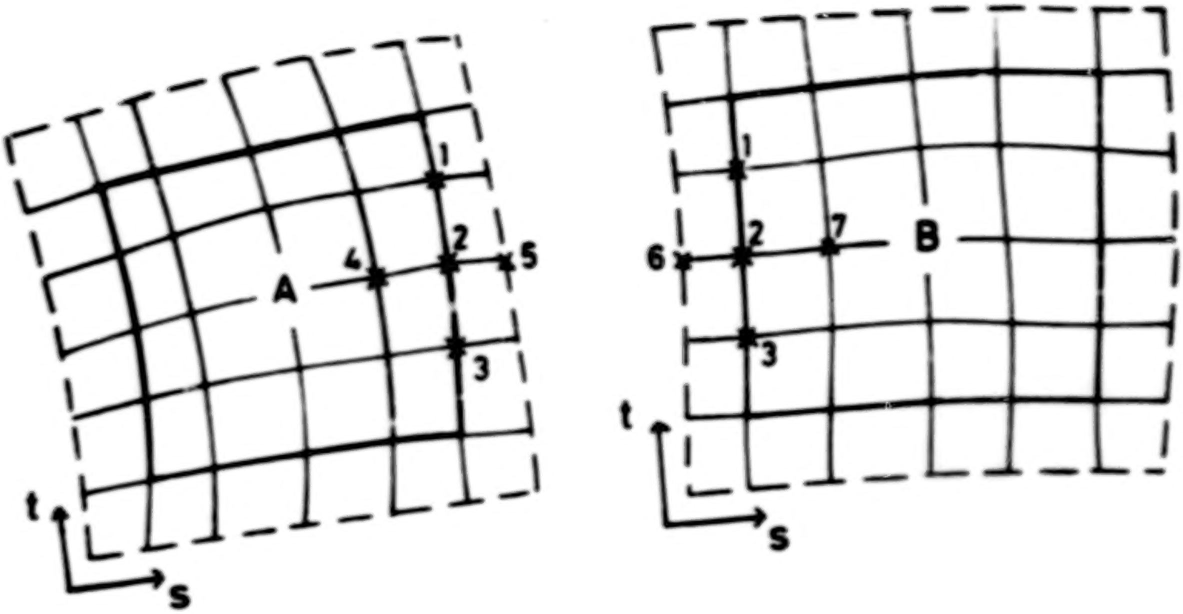
FIGURE 8 SUBSONIC PATCH BOUNDARY CONDITIONS
-DIFFERENTIAL FORM

8. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENTIAL FORM

Fundamental to the satisfactory use of patched grids is the treatment of patch boundary conditions (ie the conditions applied along the common boundary of adjacent segments). Since such boundaries do not represent real flow boundaries but simply boundaries between different local grids, the flow equation is still satisfied on these boundaries and in addition the flow velocity along and across such boundaries is continuous. These conditions are sufficient to patch the flow calculations in adjacent segments together to produce the overall flow solution. The technique is somewhat easier to apply when the flow is subsonic at the boundary points and this case will be described first.

Fig 8 shows two adjacent segments A and B where for clarity the two segments are drawn as though separated although they are actually joined along the common boundary. Also shown surrounding each segment is a row of dummy points. These points do not actually exist but are convenient for the development of the patch boundary conditions.

Taking a typical point on the common boundary the usual five point finite difference star is shown for each segment. Points 1,2,3 are common to both segments being on the common boundary. Points 4,7 are internal to segments A and B respectively while points 5,6 represent dummy points. The flow equation is solved at all internal grid points of segments A and B using ϕ on the common boundary from the previous iteration. Hence, updated values of ϕ_4 and ϕ_7 are available and it is required to calculate updated values of ϕ on the common boundary, ie ϕ_1, ϕ_2, ϕ_3.



(1) Finite difference approximation to flow equation at point 2 in segment A

$$a_1(\phi_5-2\phi_2+\phi_4)+a_2(\phi_1-2\phi_2+\phi_3)=a_6$$

(2) Finite difference approximation to flow equation at point 2 in segment B

$$b_1(\phi_7-2\phi_2+\phi_6)+b_2(\phi_1-2\phi_2+\phi_3)=b_3$$

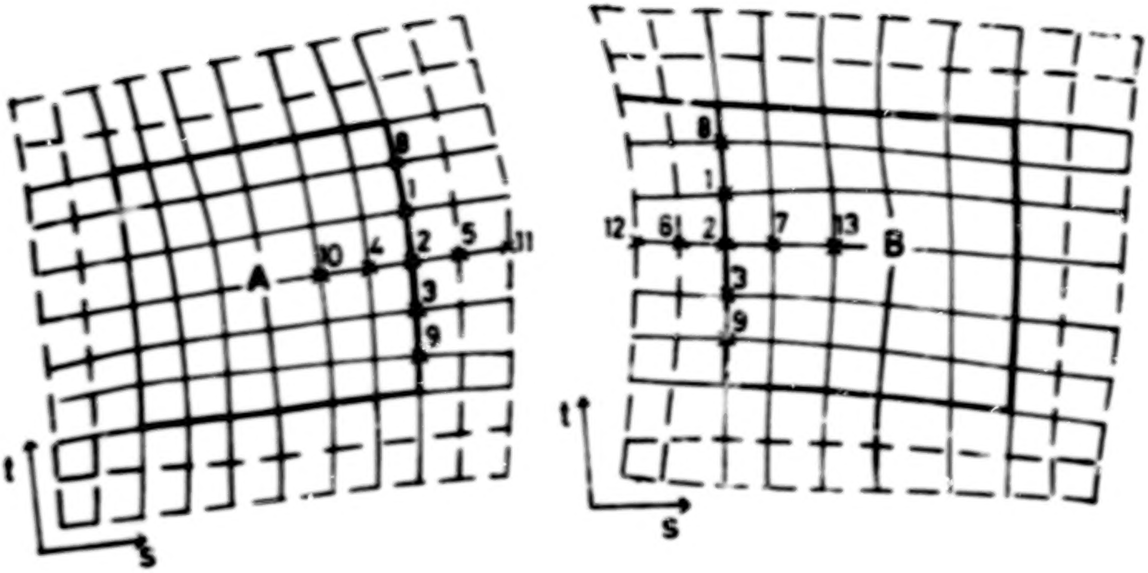
(3) Continuity of normal velocity at point 2

$$c_1(\phi_5-\phi_4)+c_2(\phi_1-\phi_3)+c_3=d_1(\phi_7-\phi_6)+d_2(\phi_1-\phi_3)+d_3$$

FIGURE 9 SUBSONIC PATCH BOUNDARY CONDITIONS DIFFERENCE FORM

9. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENCE FORM

This is done by writing finite difference approximations to the flow equation at point 2 separately for segments A and B using second order central differences which gives two equations and five unknowns (ϕ_1, ϕ_2, ϕ_3, ϕ_5, ϕ_6). A third equation with the same unknowns can be obtained using a finite difference approximation to the condition that the velocity normal to the common boundary is continuous across the boundary. Again second order central differences are used to approximate the velocities. (Continuity of velocity along the boundary is implicit in deriving the above equations). By combining these three equations the dummy values ϕ_5, ϕ_6 can be eliminated leaving one equation with three unknowns ϕ_1, ϕ_2, ϕ_3. Applying the same technique at all points along the common boundary produces a tridiagonal system of equations which may be solved for ϕ_1, ϕ_2, ϕ_3 etc using the standard algorithm.



(a) 2D flow equation in rotated form with $r = s$, $r^2 = t$

$$(c^2-q^2)v^1v^1\phi_{ss}+(c^2-q^2)v^2v^2\phi_{tt}+c^2(q^2g^{11}-v^1v^1)\phi_{ss}+c^2(q^2g^{22}-v^2v^2)\phi_{tt}$$

$$= \text{cross derivatives} + \text{low order terms}$$

(b) Continuity of normal velocity across common boundary

$$\left[\frac{v}{\sqrt{g^{11}}}\right]_A = \left[\frac{v}{\sqrt{g^{11}}}\right]$$

i.e.

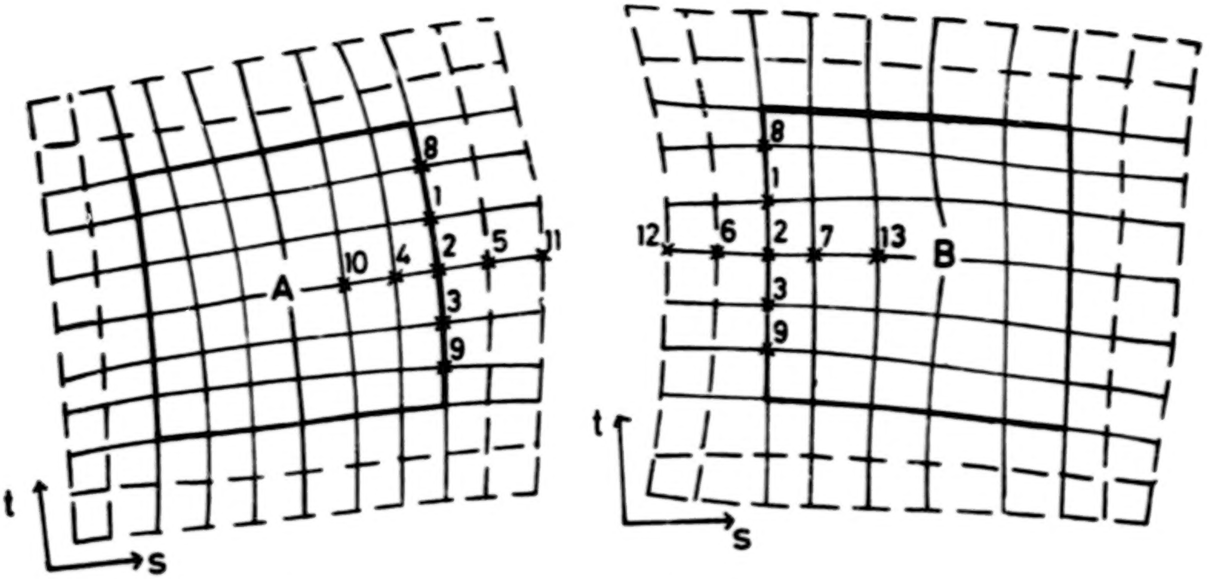
$$\left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{11}}}(\phi_t+x_t)\right]_A = \left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{12}}}(\phi_t+x_t)\right]_B$$

FIGURE 10 SUPERSONIC PATCH BOUNDARY CONDITIONS -DIFFERENTIAL FORM

10. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENTIAL FORM

When the flow at some points on a patch boundary is supersonic there is a further problem in applying the patch boundary conditions. This is due to the backward differences used to approximate some of the flow equation derivatives at supersonic points. Fig 10 again shows a pair of segments A and B patched along a common boundary. In this case, however, there are two rows of dummy points around each segment to allow for backward differencing and the difference stars have nine rather than five points.

For any specific case only seven of the nine points are actually used, which seven depending on the local flow direction. If we assume that the local flow is from bottom left to top right then in segment A points 1,2,3, 4,5,9,10 are used while points 1,2,3,6,7,9,12 are used in segment B. Comparing with the subsonic case there are now three points (4,7,10) for which updated values of ϕ are available from the solution of the flow equation at internal grid points and three dummy points (5,6,12) to be eliminated.



(1) Finite difference approximation to flow equation at point 2 in segment A

$$a_1(\phi_2 - 2\phi_4 + \phi_{10}) + a_2(\phi_2 - 2\phi_3 + \phi_9) + a_3(\phi_5 - 2\phi_2 + \phi_4) + a_4(\phi_1 - 2\phi_2 + \phi_3) = a_5$$

(2) Finite difference approximation to flow equation at point 2 in segment B

$$b_1(\phi_2 - 2\phi_6 + \phi_{12}) + b_2(\phi_2 - 2\phi_3 + \phi_9) + b_3(\phi_7 - 2\phi_2 + \phi_6) + b_4(\phi_1 - 2\phi_2 + \phi_3) = b_5$$

(3) Continuity of normal velocity at point 2 (central differenced)

$$c_1(\phi_5 - \phi_4) + c_2(\phi_1 - \phi_3) + c_3 = d_1(\phi_7 - \phi_6) + d_2(\phi_1 - \phi_3) + d_3$$

(4) Continuity of normal velocity at point 2 (backward differenced)

$$e_1(3\phi_2 - 4\phi_4 + \phi_{10}) + e_2(3\phi_2 - 4\phi_3 + \phi_9) + e_3 =$$
$$f_1(3\phi_2 - 4\phi_6 + \phi_{12}) + f_2(3\phi_2 - 4\phi_3 + \phi_9) + f_3$$

FIGURE 11 SUPERSONIC PATCH BOUNDARY CONDITIONS
-DIFFERENCE FORM

Thus four equations are required to eliminate the dummy points for the supersonic case rather than three. The first three equations are essentially the same as for the subsonic case except that appropriate second derivatives in the two approximations to the flow equation are now backward differenced rather than centrally differenced. After some experimentation we find the best equation to use for the fourth equation is another finite difference approximation to the continuity of normal velocity condition but this time approximating the velocities by second order backward differences in the usual upstream sense. This ensures that the value of ϕ at the extra dummy point (point 12 in this case) is not influenced by downstream values of ϕ which would violate the domain of dependence conditions.

When ϕ_5, ϕ_6 and ϕ_{12} have been eliminated from these four equations a single equation with four unknowns (ϕ_1, ϕ_2, ϕ_3, ϕ_9) is left. Applying the same technique at each point along the common boundary leads to a quadradiagonal system of equations which may be solved for ϕ_1, ϕ_2, ϕ_3, ϕ_9 etc. In practice, we reduce this set to a tridiagonal system, which is easier to solve, by fixing ϕ_9 at its value from the previous iteration.

Experience so far suggests that supersonic points on a patch boundary are more likely to lead to instability than are subsonic points. However, with some care it has been possible to satisfactorily compute cases with all subsonic, all supersonic and with mixed boundary points including one case where a strong shock crossed the boundary.

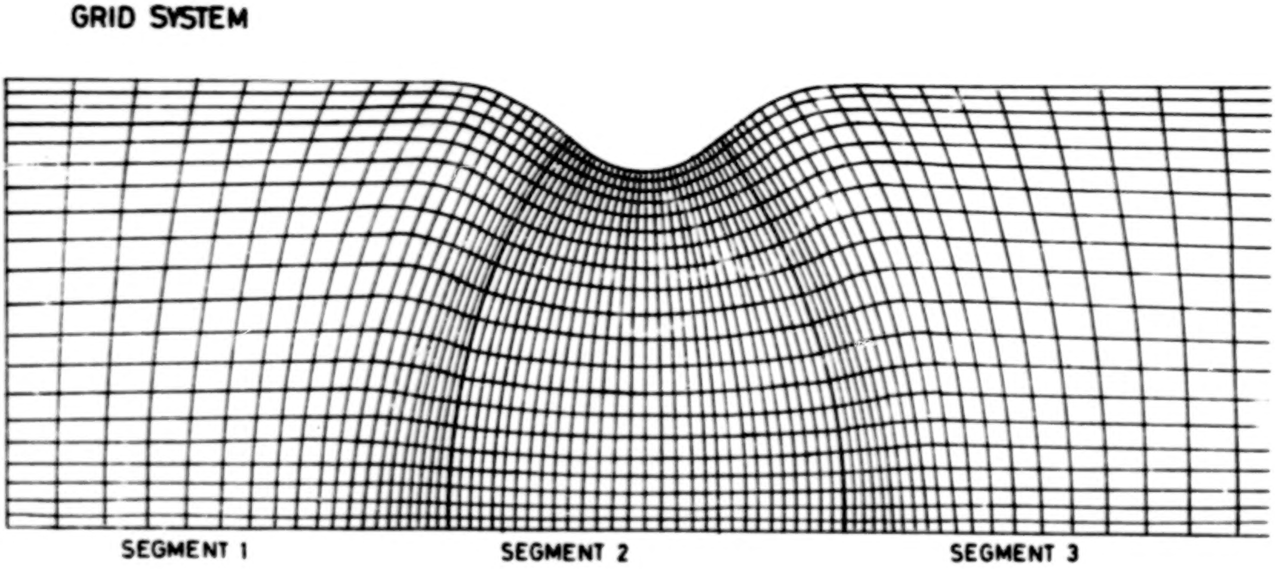

GRID SYSTEM
SEGMENT 1
SEGMENT 2
SEGMENT 3

FIGURE 12 CONDI NOZZLE

12. EXAMPLE CONDI NOZZLE GRID

In order to demonstrate the use of the method, we considered the case of a duct flow. The configuration is that of a convergent-divergent nozzle produced by a cosine distortion on the upper surface of a two-dimensional duct with an area ratio of 0.8.

Clearly, it is possible to produce a single segment system to solve this problem but for the sake of demonstration we have divided it into three segments. The interfaces between the segments are denoted by the more pronounced lines and these patches are normal to both upper and lower surfaces. Note that, although the lines appear to have continuous derivatives through the segment boundaries, this is not the case. The grid points have been distributed in an appropriate manner with a much finer grid near the bump on the upper surface. For this example the grid extends to finite distances upstream and downstream and uniform onset flow is assumed at the upstream end.

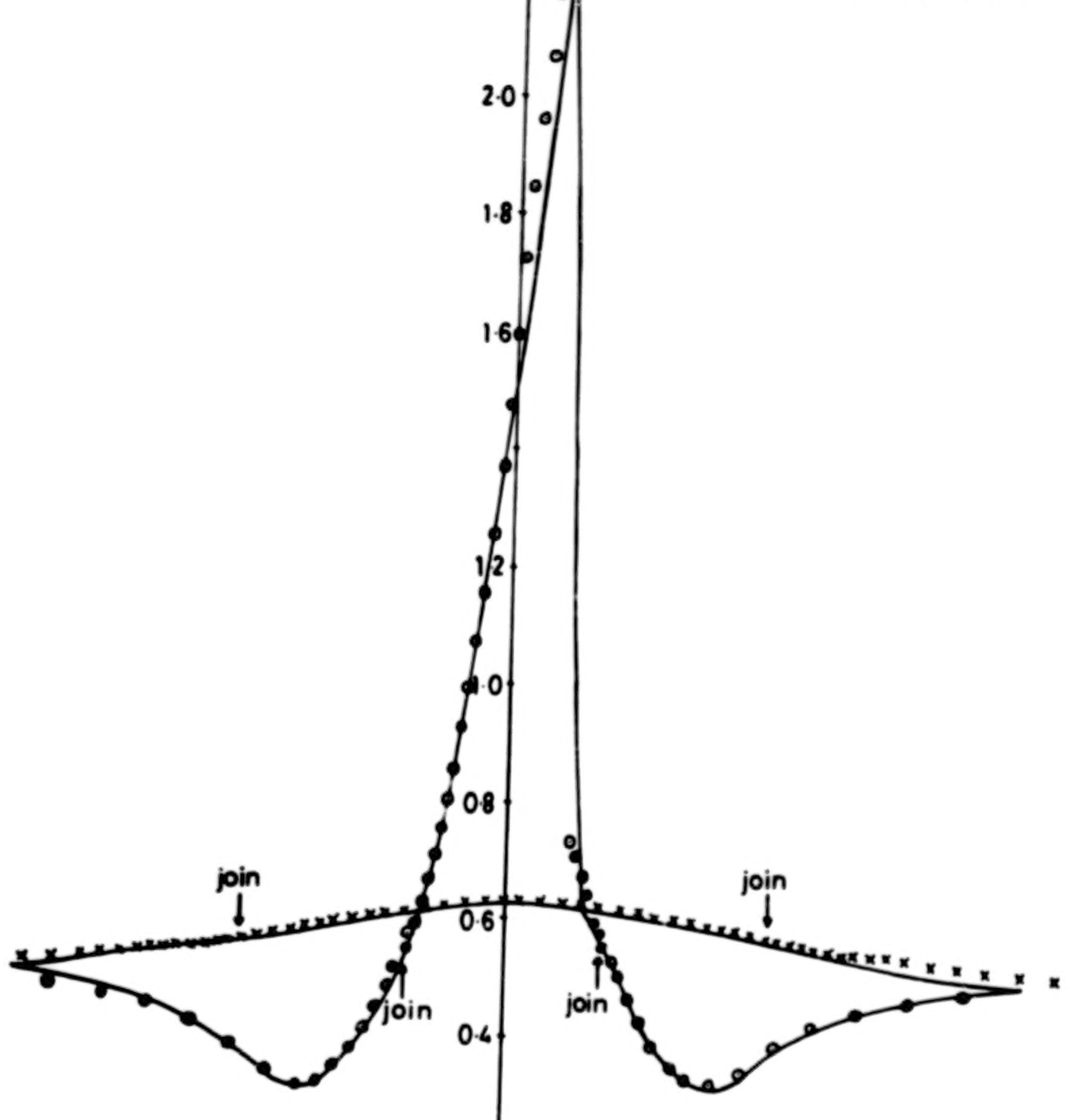


FIGURE 13 CONDI NOZZLE

13. EXAMPLE CONDI NOZZLE RESULTS

In order to check the results, comparisons have been made with a well established nozzle program developed by Baker at ARA. The case shown here is for an onset Mach number of 0.5. The inviscid flow solution, which is outside the range of validity of a potential method, is nevertheless an appropriate test case with a very strong shock. The agreement between the two methods is very encouraging and the patching does not appear to have affected the solution. However, possibly due to slow convergence, there was a small discontinuity across the patch but when the mean value is used, the result is reasonably smooth. The locations of the segment interfaces are shown on the figure. Although they are not shown here, changes in the position and number of patches did not affect the result significantly.

CONCLUSIONS AND PROPOSALS FOR FURTHER WORK

1 Initial use of grid patching is encouraging

2 The range of application has been limited

3 Further cases are now being attempted

e.g. aerofoil in wind tunnel

cascade flows

intake flows

4 Modifications required for unrestricted far field

5 Extension to three dimensions

FIGURE 14

14. CONCLUSIONS AND PROPOSALS FOR FURTHER WORK

The grid patching technique has been investigated using two-dimensional test cases and initial results are encouraging. However, the range of application has, so far, been limited and cases with a greater number of segments are now being attempted. These include an aerofoil in a wind tunnel together with the two configurations shown earlier, cascade and intake flows. For the latter case some work is required in introducing extra transformations for an unrestricted far field.

We should then be in a position to deal with most two-dimensional problems and this should form the basis of extending the techniques into three dimensions.

BLANK PAGE

BLANK PAGE

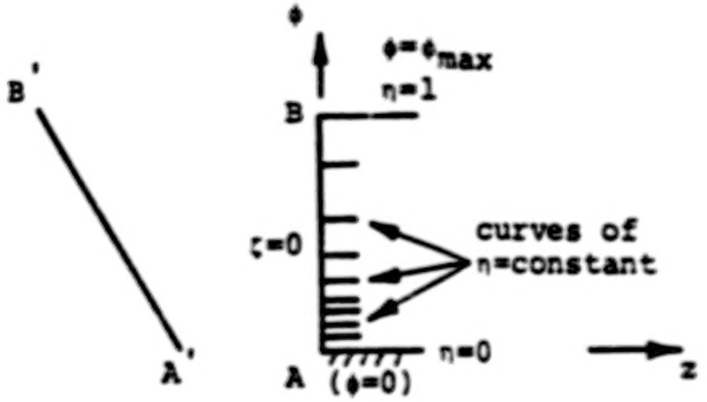
BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES

U. GHIA AND K.N. GHIA

UNIVERSITY OF CINCINNATI, CINCINNATI, OHIO

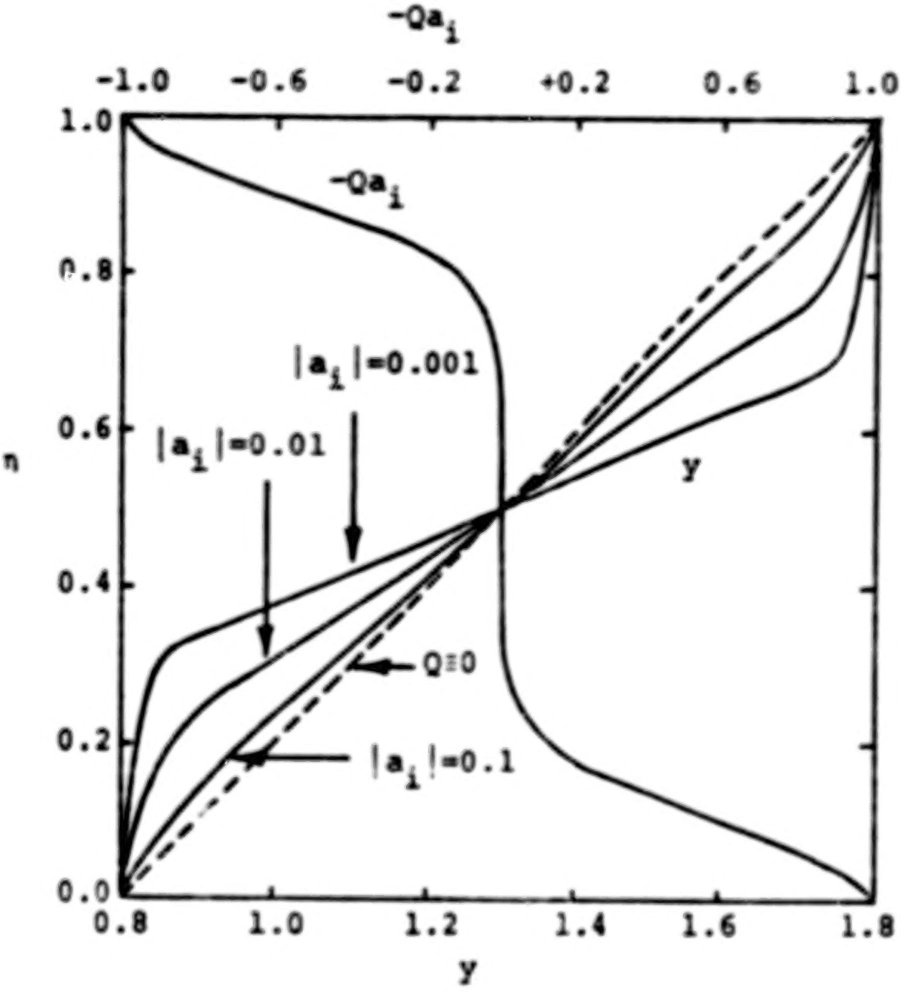
A procedure has been developed, using the differential-equation approach, for generating boundary-fitted coordinates for regions with highly curved boundaries as well as reentrant boundaries, such as those encountered in breaking surface waves. The resulting coordinates are nearly orthogonal and can provide adequate resolution even in the reentrant region. Consistent treatment of end boundaries and the use of a systematic initialization scheme and advanced implicit numerical solution techniques make the procedure highly efficient. The method developed for implicit enforcement of the periodicity boundary condition should be beneficial in the analysis of turbomachinery flow applications.

CONSISTENT TREATMENT OF END-BOUNDARIES



A limiting form of the coordinate equations at the end-boundary is solved to determine, prior to the complete solution, the point distribution at this boundary, consistent with the interior distribution. This procedure avoids discontinuities in the transformed-coordinate derivatives near the end-boundaries, while maintaining Dirichlet boundary conditions for the transformation.

SOLUTION OF LIMITING EQUATION AT END-BOUNDARY



$$\phi_{\eta\eta} + Q\,\phi_\eta^3 = 0$$

where

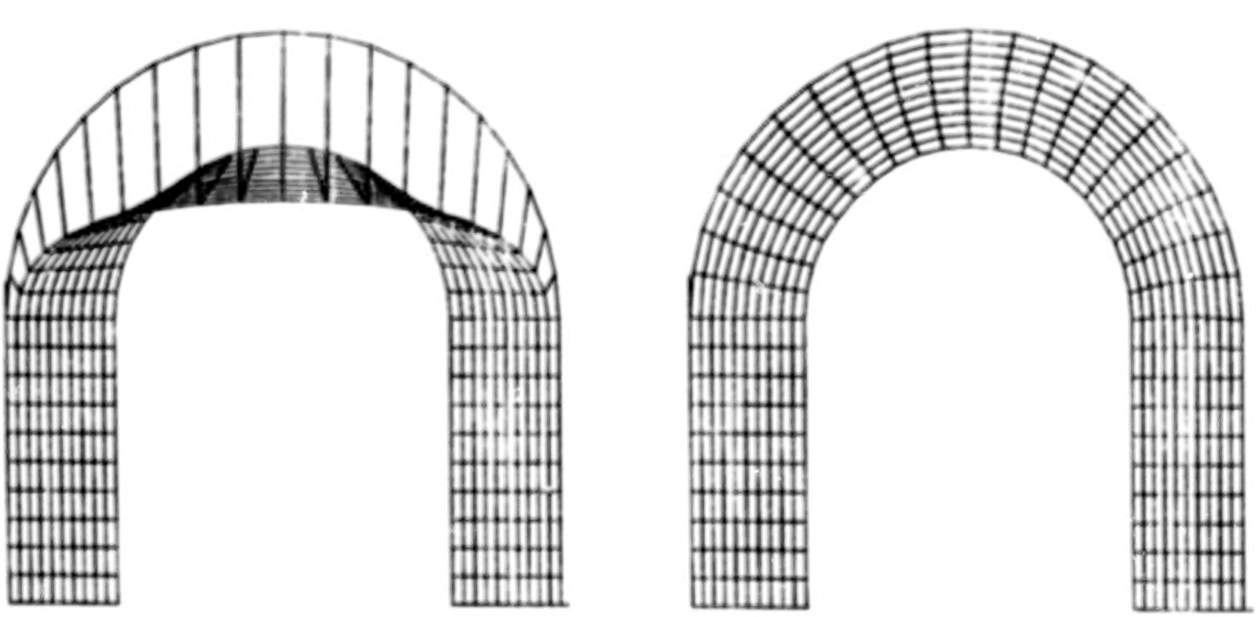
$$Q(\eta) = \sum_{k=1}^{2} \frac{1}{a_k} \exp[-(\eta-\eta_k)^2/(2b_k^2)]$$

$$a_1 < 0 \;, \qquad |a_1| = a_2$$

$$b_1 = b_2 = 0.1$$

$$\eta_1 = 0 \;, \qquad \eta_2 = 1$$

INITIALIZATION PROCEDURE

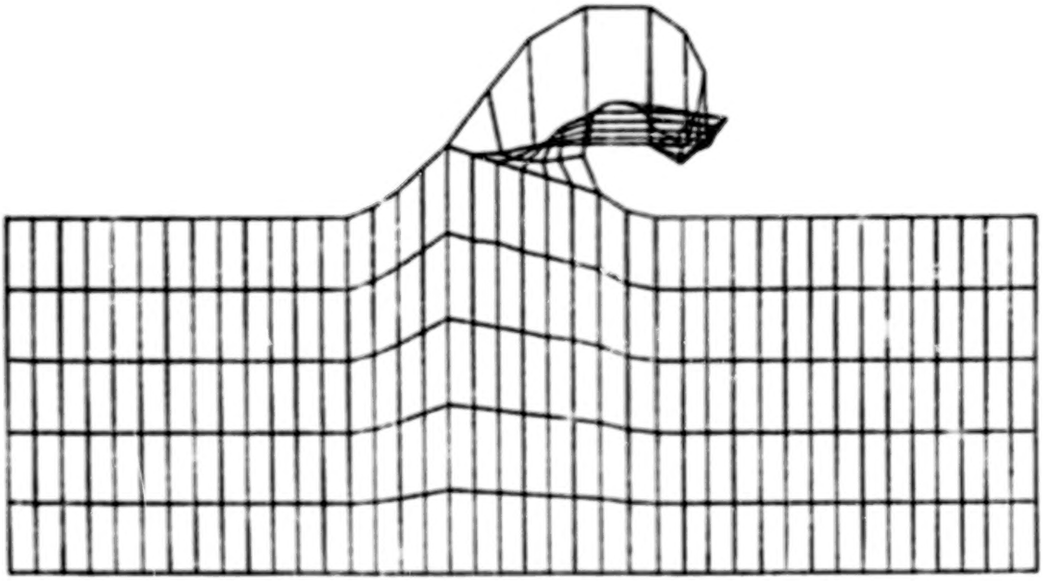


GEOMETRIC INITIALIZATION

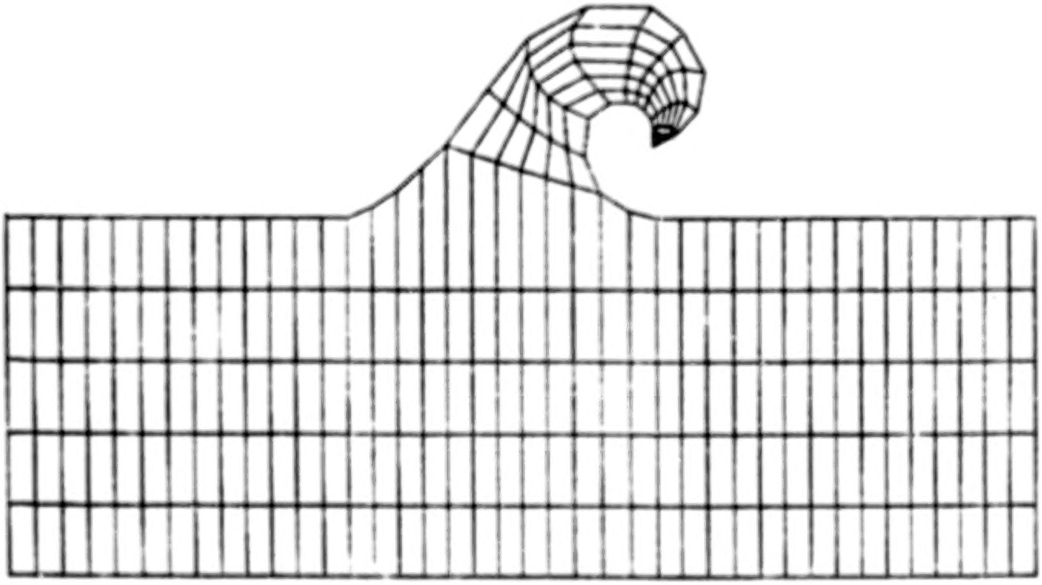
INITIALIZATION BY LOCALLY SELF-SIMILAR SOLUTION

INITIALIZATION PROCEDURE

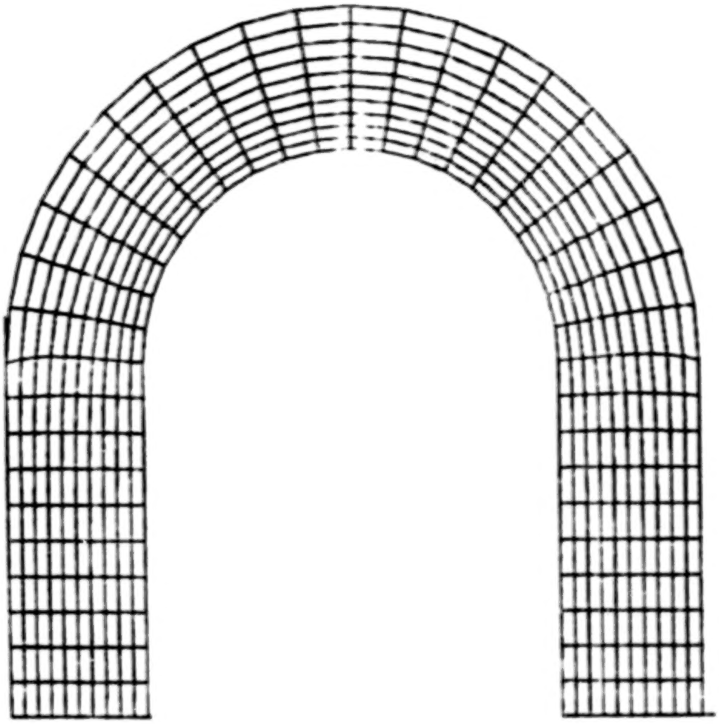
GEOMETRIC INITIALIZATION



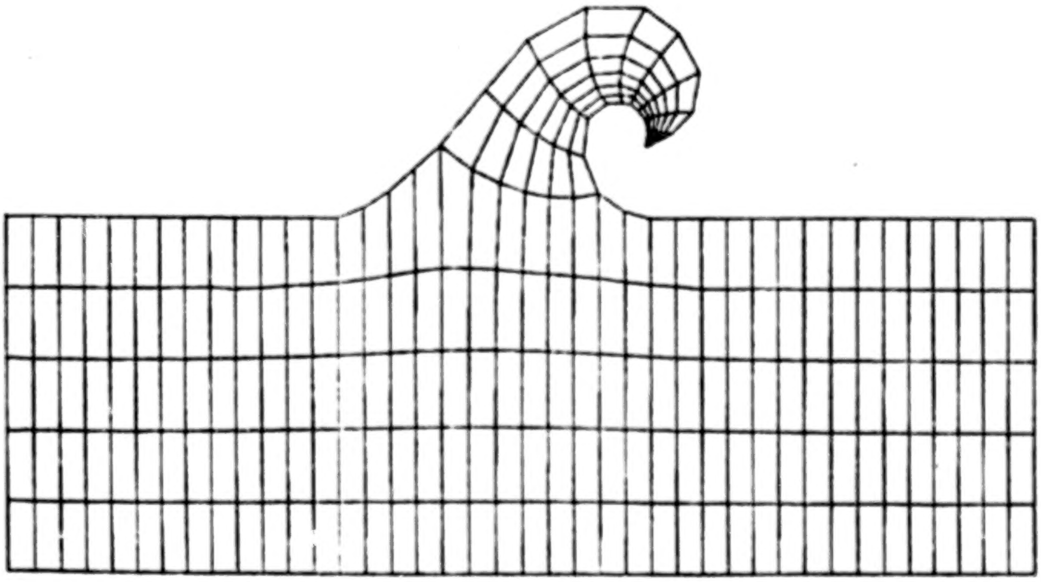
INITIALIZATION BY LOCALLY SELF-SIMILAR SOLUTION



SURFACE-ORIENTED COORDINATES FOR DUCT WITH HIGHLY CURVED BOUNDARIES



BOUNDARY-ORIENTED COORDINATES FOR A TYPICAL SURFACE WAVE WITH REENTRANT BOUNDARIES



IMPLICIT ENFORCEMENT OF PERIODICITY BOUNDARY CONDITION

DIFFERENTIAL EQUATION:

$$\phi'' + a\phi = b$$

PERIODICITY BOUNDARY CONDITIONS:

$$\phi_0 = \phi_1 = A \; ; \qquad \phi_0' = \phi_1' = B$$

where A and B are unknown.

SOLUTION PROCEDURE: Let $\phi = Af + Bg + h$

$$f'' + af = 0 \qquad g'' + ag = 0 \qquad h'' + ah = b$$

with

$$f_0 = 1 \qquad g_0 = 0 \qquad h_0 = 0$$

$$f_1' = 0 \qquad g_1' = 1 \qquad h_1' = 0$$

Then,

$$Af_1 + Bg_1 + h_1 = A$$

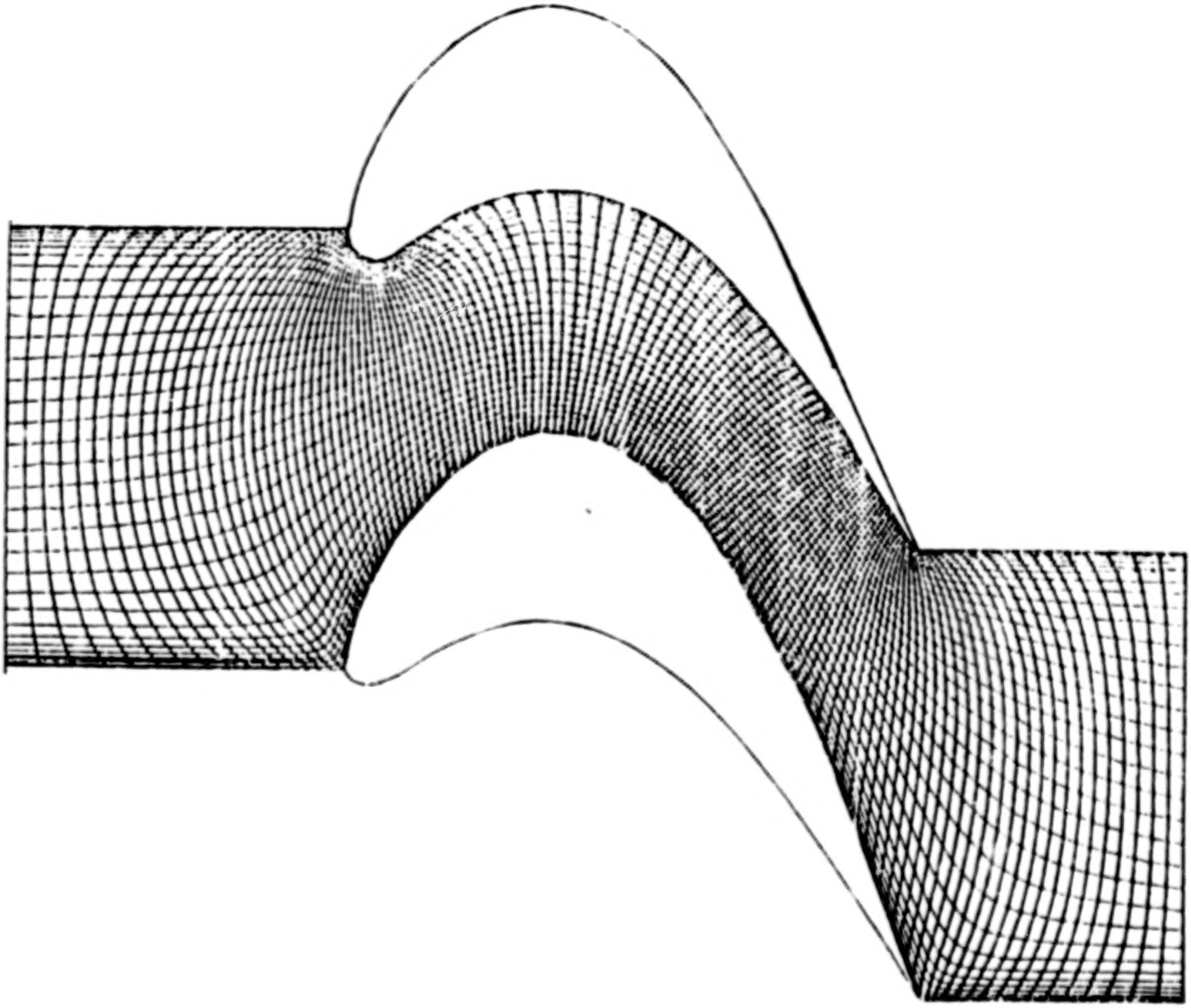
$$Af_0' + Bg_0' + h_0' = B$$

so that

$$A = [h_1(1-g_0') + h_0' \, g_1]/[(1-f_1)(1-g_0') - f_0' \, g_1]$$

$$B = [h_0'(1-f_1) + f_0' \, h_1]/[(1-f_1)(1-g_0') - f_0' \, g_1]$$

SURFACE-ORIENTED COORDINATES FOR A TURBINE CASCADE - (129 x 33) NONUNIFORM GRID WITH EASILY APPLICABLE PERIODICITY



STREAMWISE-ALIGNED SURFACE-ORIENTED COORDINATES FOR A TYPICAL TURBINE CASCADE - (161 x 33) NONUNIFORM GRID

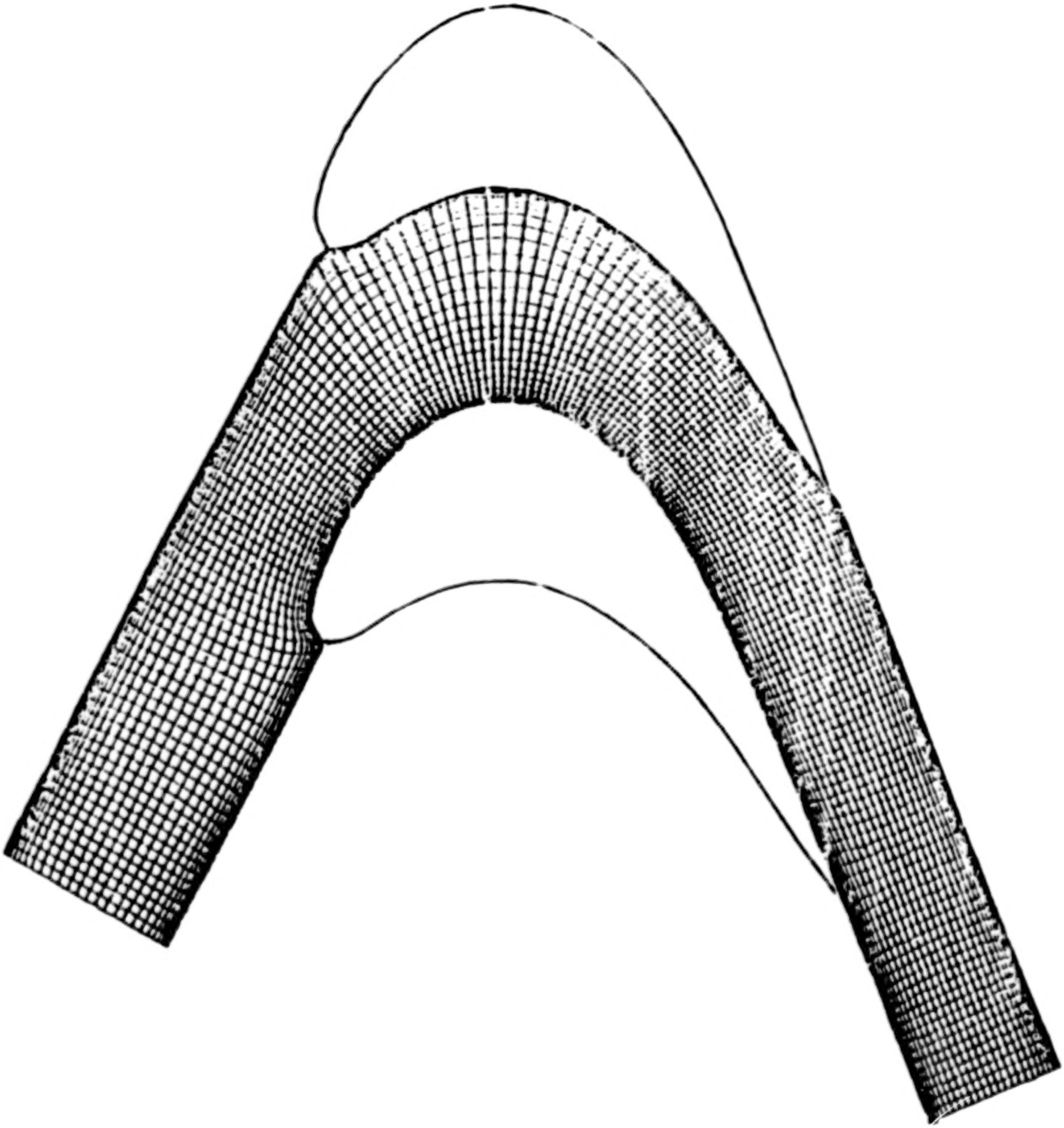


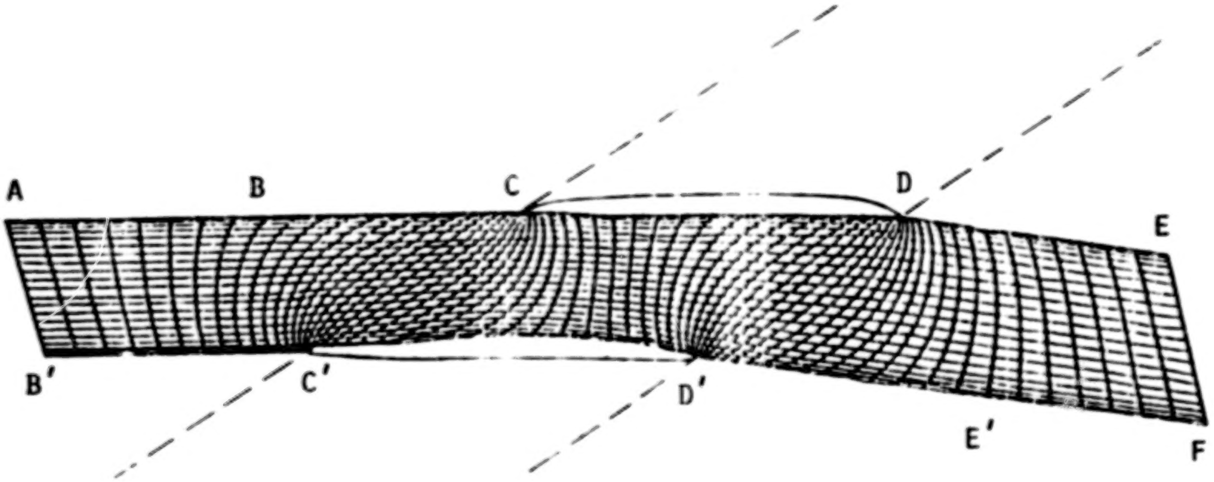
TABLE 1. EFFECT OF MULTIGRID (MG) ITERATION TECHNIQUE ON CONVERGENCE OF COORDINATE SOLUTION FOR CASCADE WITH EASILY APPLICABLE PERIODICITY

Method	Grid	Work Units of Resp. Finest Grid	CPU Seconds	Remarks
ADI	(65 x 17)	100	37.69	uniform spacing
SIP	(65 x 17)	53	11.96	uniform spacing
MG-SIP	(65 x 17)	6.5	2.08	uniform spacing
ADI	(65 x 17)	95	36.67	nonuniform spacing
SIP	(65 x 17)	25	6.33	nonuniform spacing
MG-SIP	(65 x 17)	7.5	2.32	nonuniform spacing
MG-SIP	(129 x 33)	6.4	8.44	nonuniform spacing

TABLE 2. CONVERGENCE OF COORDINATE SOLUTION FOR CASCADE GEOMETRY WITH PERIODICITY USING A STRONGLY IMPLICIT PROCEDURE (SIP) AND MULTIGRID (MG) TECHNIQUE

Method	Grid	Work Units of Resp. Finest Grid	CPU Seconds	Remarks
SIP	(161 x 33)	81.00	≈100.0	uniform spacing. convergence is one order less than for nonuniform spacing.
MG-SIP	(161 x 33)	7.48	10.79	uniform spacing
MG-SIP	(161 x 33)	8.23	11.49	nonuniform spacing
MG-SIP	(81 x 17)	8.88	4.02	nonuniform spacing

TYPICAL SURFACE-ORIENTED COORDINATES FOR A CASCADE WITH HIGH STAGGER - (65 x 21) NONUNIFORM GRID



This figure shows a multiple-circular-arc supersonic compressor cascade with a large stagger angle and a typical coordinate distribution for such a cascade. The grid lines are concentrated near the surface of both the blades, especially near their leading and trailing edges, in order to provide good resolution for the viscous and shock effects in these regions. In addition to the nonuniform distribution of the grid points, an effort has been made to maintain near-orthogonality wherever possible. The existing non-orthogonality can be easily removed by increasing the number of points in the streamwise direction, although the coordinate distribution shown in this figure may actually be preferred for supersonic cascades. Moreover, the point distribution along the free boundaries is such as to enable enforcement of the periodicity condition, i.e., the point distributions along BC and DE are the same as along B'C' and D'E', respectively. The number of working units required to generate the (65 x 21) coordinates shown was 8.44 using the SIP-multigrid method; the corresponding CPU time was 3.48 seconds.

CONCLUSIONS

- Generation of coordinates for regions with highly curved boundaries requires suitable initial conditions; locally self-similar equations provide an excellent non-iterative initial solution.
- Generation of appropriate Dirichlet boundary conditions even with non-zero forcing functions enhances solution convergence rate.
- Use of implicit numerical solution procedures together with the multigrid iteration technique constitutes an effective method for solution of the nonlinear governing differential equations with large number of grid points.
- An adaptive coordinate distribution is formulated for the breaking surface-wave problem with a reentrant boundary; solutions are presently being obtained for a free surface wave starting from an initial sinusoidal form and undergoing the breaking phenomenon.

A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS

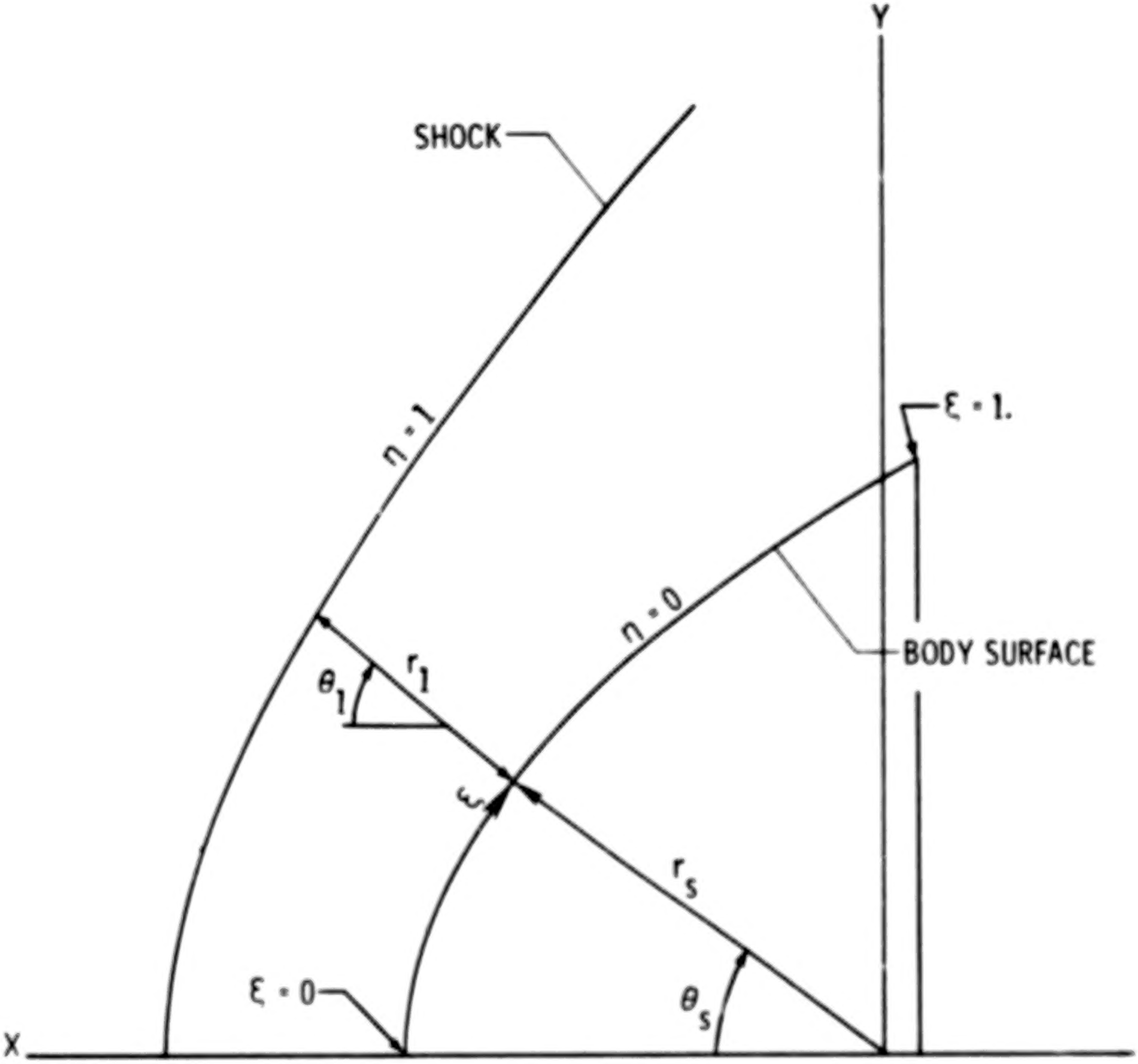
Randolph A. Graves, Jr.
OAST Aerodynamics Office
NASA Headquarters
Washington, DC

Abstract

An application of a simple numerical technique which allows for the rapid construction of orthogonal coordinate systems about two dimensional and axisymmetric bodies is presented. This technique which is based on a "predictor-corrector" numerical method is both simple in concept and easy to program. It can be used to generate orthogonal meshes which have unequally spaced points in two directions. These orthogonal meshes in their transformed computational plane are, however, equally spaced so that the differencing for the metric coefficients and the fluid dynamic equation terms can be easily determined using equally spaced central finite differences. Solutions to the Navier-Stokes equations for flow over blunt bodies with reverse curvature are presented. The coupling of the time dependent fluid dynamic equations and the coordinate generator worked well with no undersirable effects noted.

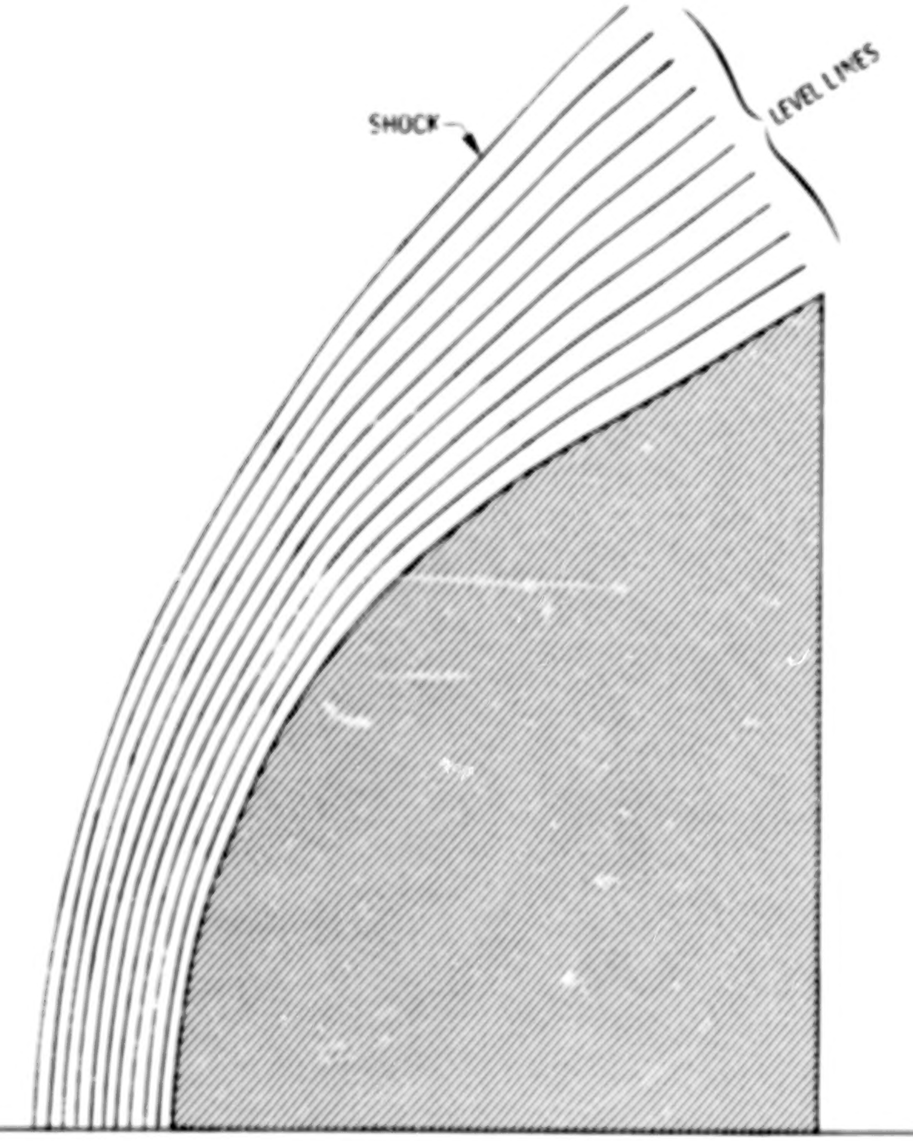
Flowfield Geometrical Relationships

The numerically generated orthogonal coordinates are determined from the original cartesian coordinate systems description of the body surface and outer boundary. Taking the origin of the X,Y system as lying inside the body to be described, the surface distance ξ, which forms one of the transformed orthogonal coordinates, can be easily calculated by defining ξ as zero at origin of the region of interest and increasing to unity at the end of the region (nondimensionalized surface distance). The other orthogonal coordinate, η, is taken as zero on the body surface and as unity on the outer boundary. Thus the region of interest is transformed into a nondimensional square.



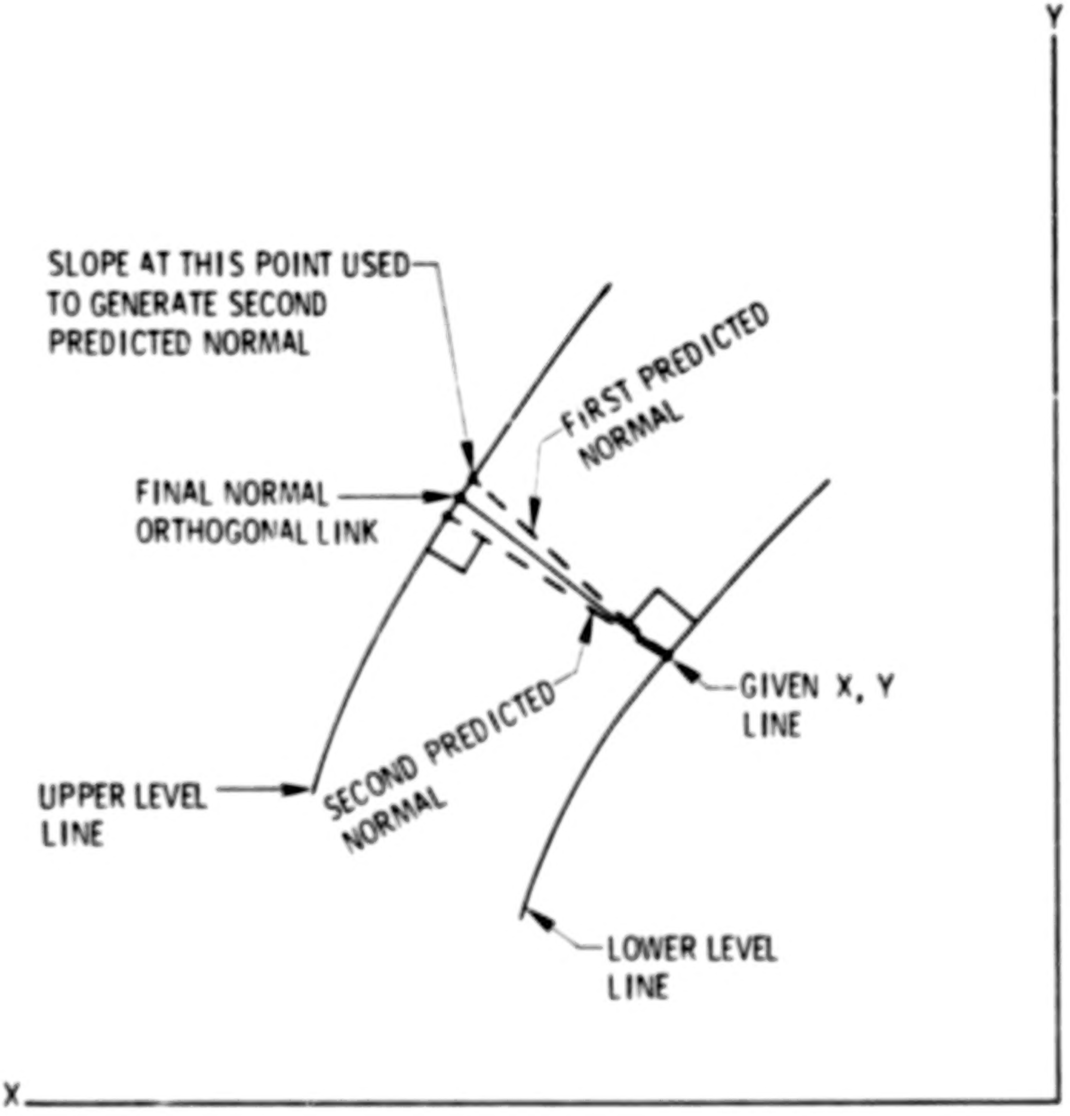
Level Line Construction

The level lines between the outer boundary and the body surface can be constructed arbitrarily; however, the easiest approach is to construct the level lines along straight lines connecting corresponding points on the body and the outer boundary. The mesh points on the outer boundary are not the final mesh points but initial values used only to set up the level lines. The actual mesh points will result from the numerical generation of the orthogonal normal lines. The spacing of the level lines is arbitrary and highly stretched meshes can be easily constructed.



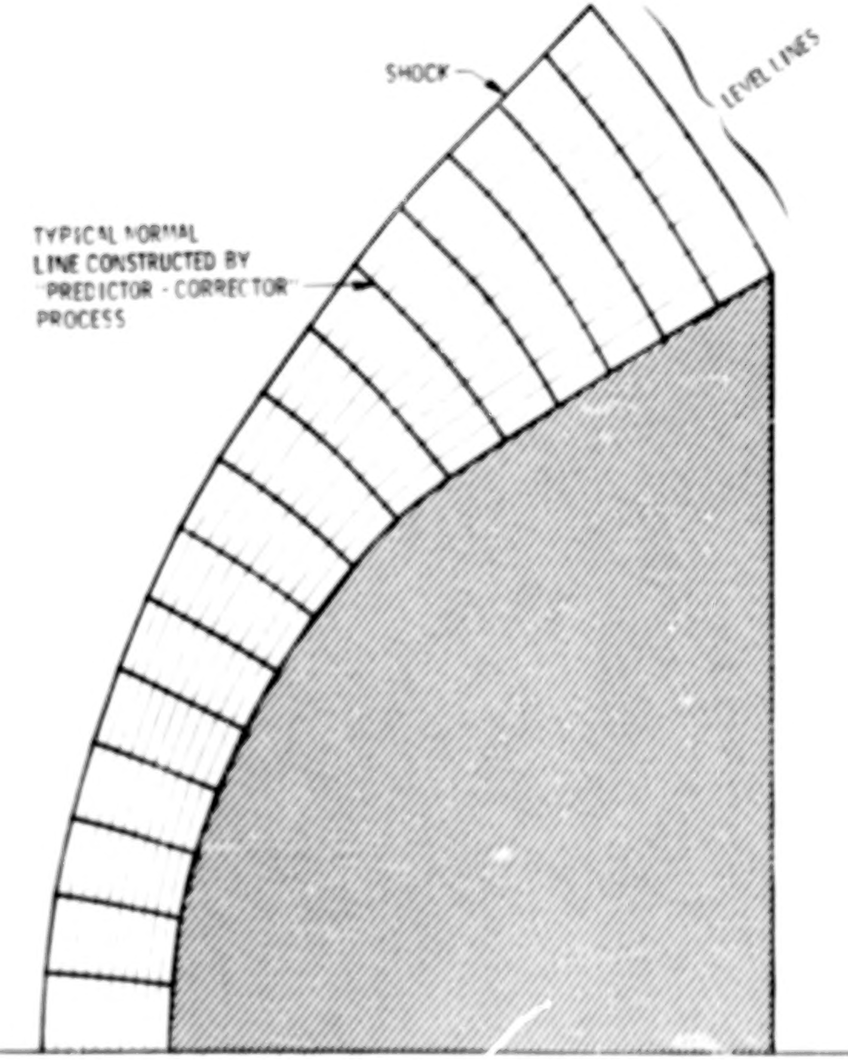
Normal Line Construction Technique

Once the level lines have been determined, the normal lines are constructed numerically so that an orthogonal system is defined. The approach to the construction of the normal lines is the one given by McNally which uses a simple "predictor-corrector" technique analogous to the trapezoidal integration method of numerical integration. In this technique, the solution is first predicted from the level line at a known point by using the Euler method. Once the predicted point on the next level line is obtained, the slope at that point is calculated and a new predicted point is obtained using this slope. The actual solution is then a combination of these two solutions, i.e. the final X,Y values are an average of the predicted and corrected ones.


Y
SLOPE AT THIS POINT USED TO GENERATE SECOND PREDICTED NORMAL
FIRST PREDICTED NORMAL
FINAL NORMAL ORTHOGONAL LINK
GIVEN X, Y LINE
SECOND PREDICTED NORMAL
UPPER LEVEL LINE
LOWER LEVEL LINE
X

Typical Coordinate Mesh Construction

Starting on the body, the normal line construction technique proceeds point by point along a level line until all normals on that level have been constructed. The solution then proceeds to the next level and the process is continued until the outer boundary is reached. Thus the complete mesh system is numerically generated in a simple straight forward, noniterative process. Since the computational plane (ξ, η) is an equally spaced rectangular region, the metric coefficients can be determined from the completed mesh system using equally spaced finite difference relations. Fourth order accurate difference relations are recommended as they provide for smoothly varying metric coefficients.

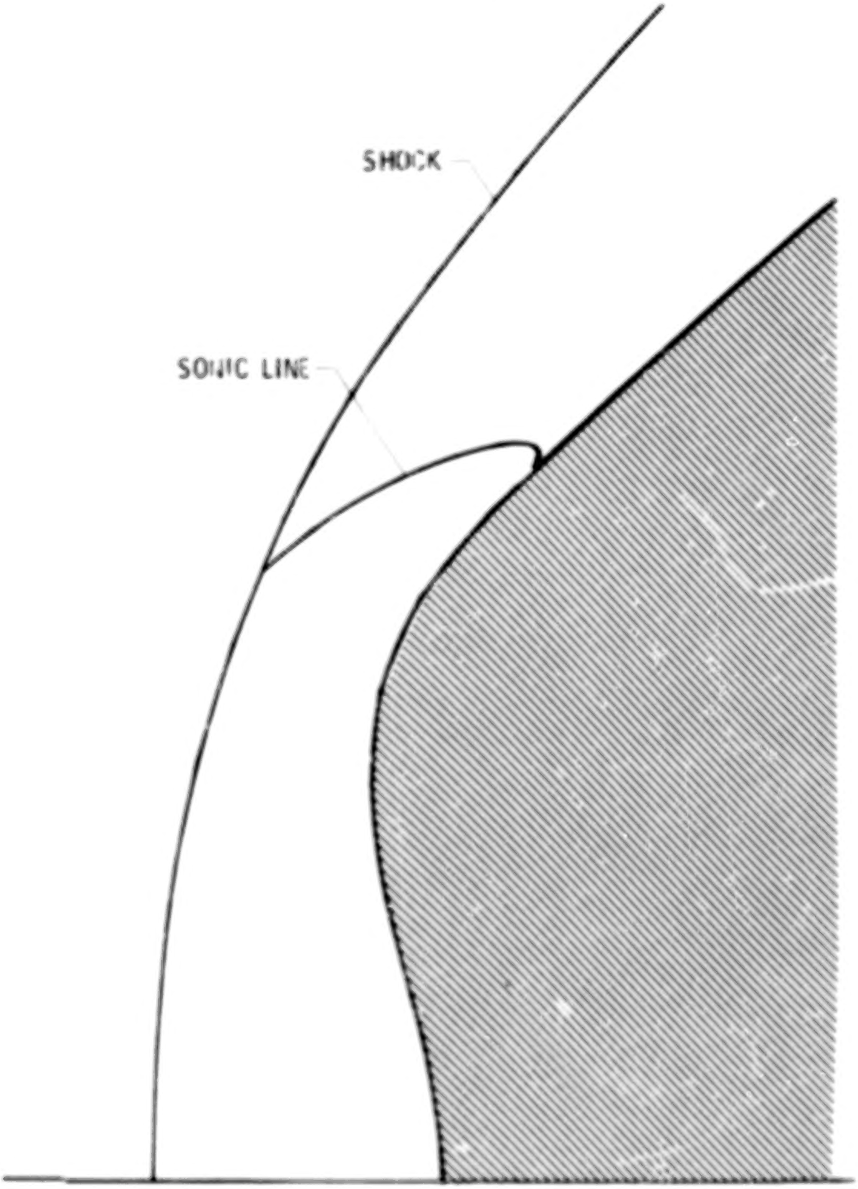

SHOCK
LEVEL LINES
TYPICAL NORMAL LINE CONSTRUCTED BY "PREDICTOR - CORRECTOR" PROCESS

Shock and Sonic Line

Solutions to the laminar flow Navier-Stokes equations were obtained for flow over bodies with blunted noses, including reverse curvature. These bodies were generated using the following cubic forebody generator,

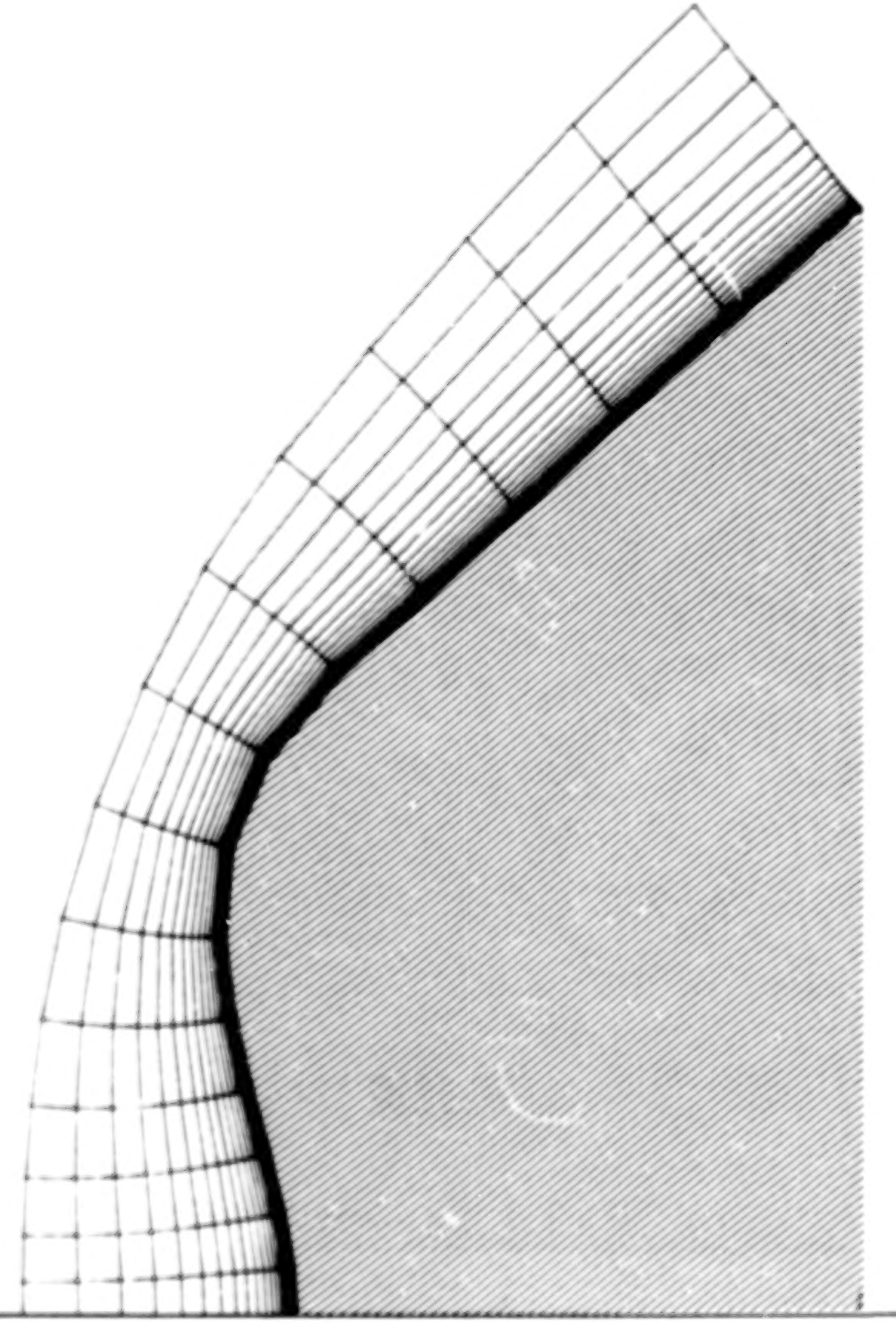
$$X=X_o + A_1 y^2 + A_2 y^3$$

where X_o determines the nose offset while the coefficients A_1 and A_2 are determined such that the forebody nose section joins smoothly to the conical flank. This solution was run for a free stream Mach number of 10.33 and X_o=.4. The shock shape and sonic line are typical of the solution for bodies with very blunt nose regions.



Converged Coordinate System

The converged coordinate system shown for $X_o=.4$ is composed of 15 transverse stations and 31 normal stations. The normal direction spacing is highly stretched to provide resolution for the boundary layer. There is only mild stretching in the transverse direction to provide for improve stagnation region resolution. There were no undesirable effects noted in the coupling of the viscous flow calculations with the coordinate generation.



BLANK PAGE

BLANK PAGE

A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS

DARRYL W. HALL
SCIENCE APPLICATIONS, INC.
McLEAN, VIRGINIA

ABSTRACT

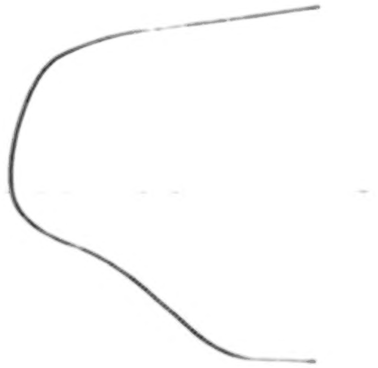
This presentation describes a three-dimensional body-fitted coordinate system developed for use in the calculation of inviscid flows over ablated, asymmetric reentry vehicle nosetips. Because of the potential geometric asymmetries, no standard coordinate system (e.g., spherical, axisymmetric reference surface-normal) is capable of being closely aligned with the nosetip surface. To generate a 3-D, body-fitted coordinate system an analytic mapping procedure is applied that is conformal within each meridional plane of the nosetip; these transformations are then coupled circumferentially to yield a three-dimensional coordinate system. The mappings used are defined in terms of "hinge points", which are points selected to approximate the body contours in each meridional plane. The selection of appropriate hinge points has been automated to facilitate the use of the resulting nosetip flow field code.

PROBLEM DEFINITION

The goal of this effort is the development of a procedure for calculating supersonic/hypersonic inviscid flows over asymmetric ablated reentry vehicle nosetips. These asymmetric shapes, such as illustrated in this figure, result from asymmetric transition on the nosetip, which occurs at the lower altitudes during reentry (i.e., below 15.24 km). Because these shapes occur in the high Reynolds number, turbulent regime, with thin boundary layers, an inviscid solution is capable of accurately predicting the pressure forces on the nosetip. The nosetip flow field solution is also required to provide the required initial data for afterbody calculations; this coupling of nosetip and afterbody codes allows accurate prediction of the effects of the nosetip shape on the afterbody flow field.

The flow field code developed is a finite-difference solution of the unsteady Euler equations in "non-conservation" form (i.e., the dependent variables are the logarithm of pressure, P, the velocity components, u,v,w, and the entropy, s). In this approach the steady flow solution is sought as the asymptotic limit of an unsteady flow, starting from an assumed initial flow field.

CALCULATION OF SUPERSONIC/HYPERSONIC INVISCID FLOWS OVER ASYMMETRIC ABLATED REENTRY VEHICLE NOSETIPS



ASYMMETRIC ABLATED NOSETIP SHAPE

APPROACH

- FINITE-DIFFERENCE SOLUTION OF UNSTEADY EULER EQUATIONS
- STEADY FLOW SOLUTION SOUGHT AS THE ASYMPTOTIC LIMIT OF UNSTEADY FLOW

COORDINATE SYSTEM REQUIREMENTS

It is well known that accurate numerical calculation of fluid flows requires the use of a coordinate system closely aligned with the principal features of the flow. For the nosetip problem this requirement would be satisfied by a coordinate system which closely follows the body shape and, hence, the streamlines of the flow. Because of the asymmetric nosetip geometries being considered, standard coordinate systems (e.g., spherical, axisymmetric reference surface-normal) are incapable of being aligned with the nosetip surface at all points. Thus, a coordinate transformation is sought that will align the coordinate system with an arbitrary nosetip geometry. By requiring the transformation to be in analytic form, the need of solving partial differential equations to define the transformation can be avoided. Finally, the transformation should be in a form that readily lends itself to automated definition, minimizing the inputs required of a user of the code.

OPTIMUM COORDINATE SYSTEM FOR NUMERICAL FLOW FIELD CALCULATIONS IS BODY-ORIENTED

COORDINATE TRANSFORMATION SOUGHT THAT:

1.) ALIGNS COORDINATE SURFACES WITH BODY SURFACE

2.) IS ANALYTIC (SOLUTION OF PDE'S NOT REQUIRED TO DEFINE TRANSFORMATION)

3.) CAN BE READILY AUTOMATED (TO MINIMIZE INPUTS REQUIRED FROM USER)

COORDINATE TRANSFORMATION

The nosetip geometry is defined in an (x,y,ϕ) cylindrical coordinate system, and a mapping to a (ξ,η,θ) transformed coordinate system is sought. Since current reentry vehicle nosetips are initially axisymmetric (prior to ablative shape change), it is assumed that nosetip cross-sections retain some "axisymmetric" character during reentry. Thus, no transformation of the circumferential coordinate is required, and $\theta = \phi$ is assigned. (This transformation can readily be generalized to $\theta = f(\phi)$ if required for other applications of this approach.) Within a ϕ = constant meridional plane, the transformation reduces to the two-dimensional form $\xi = \xi(x,y)$, $\eta = \eta(x,y)$. Conformal transformations from the $z = x+iy$ to the $\zeta = \xi+i\eta$ plane are desirable, ensuring that an orthogonal (ξ,η) grid maps back onto an orthogonal grid in the (x,y) plane.

(x,y,ϕ) CYLINDRICAL COORDINATES IN PHYSICAL SPACE

(ξ,η,θ) COORDINATES IN TRANSFORMED SPACE

TRANSFORMATION OF CIRCUMFERENTIAL COORDINATE NOT REQUIRED (NOSETIPS INITIALLY AXISYMMETRIC); ASSUME TRANSFORMATION TAKES THE FORM

$$\xi = \xi\,(x,y,\phi)$$

$$\eta = \eta\,(x,y,\phi)$$

$$\theta = \phi$$

IN A MERIDIONAL PLANE (ϕ = CONSTANT), THE TRANSFORMATION REDUCES TO

$$\xi = \xi\,(x,y)$$

$$\eta = \eta\,(x,y)$$

DEFINITION OF TRANSFORMATION

The approach used to define the coordinate transformations is a modification of the "hinge point" approach of Moretti*. The mapping is defined as a sequence of conformal transformations of the form

$$z_{j+1} - 1 = [z_j - h_{j+1,j}]^{\delta_j}$$

where $z_j = x_j + iy_j$ ($j = 1$ is physical space) and $h_{i,j}$ is the $i^{\underline{th}}$ hinge point in the z_j plane. The hinge points in the physical (z_1) plane are selected to approximately model the body geometry. By mapping the hinge points sequentially onto the horizontal axis, the image of the body surface will then be a nearly horizontal contour.

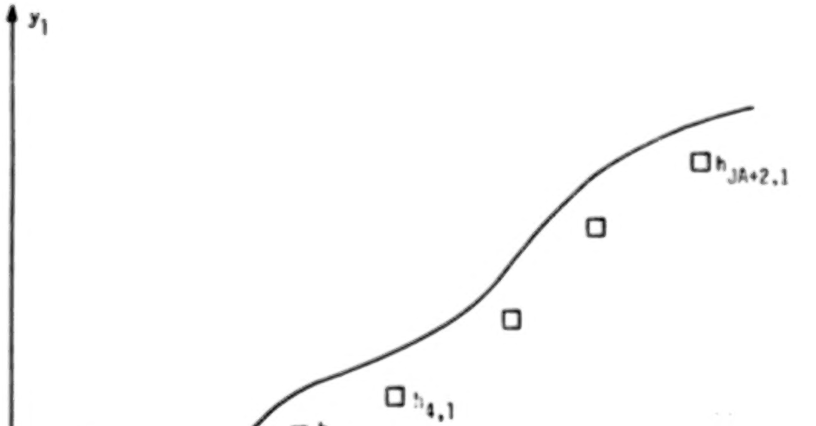
INDEPENDENTLY IN EACH MERIDIONAL PLANE, DEFINE A SEQUENCE OF CONFORMAL TRANSFORMATIONS

$$z_{j+1} - 1 = [z_j - h_{j+1,j}]^{\delta_j} \quad j = 1,2,\ldots,JA$$

$$z_j = x_j + i_{yj} \text{ (j = 1 IS PHYSICAL SPACE)}$$

$$h_{i,j} = i^{\underline{th}} \text{ "HINGE POINT" IN } j^{\underline{th}} \text{ SPACE}$$

HINGE POINTS ARE SELECTED TO APPROXIMATE BODY GEOMETRY



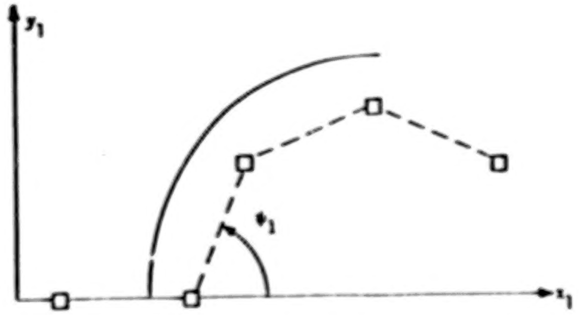
HINGE POINT DEFINITION

---

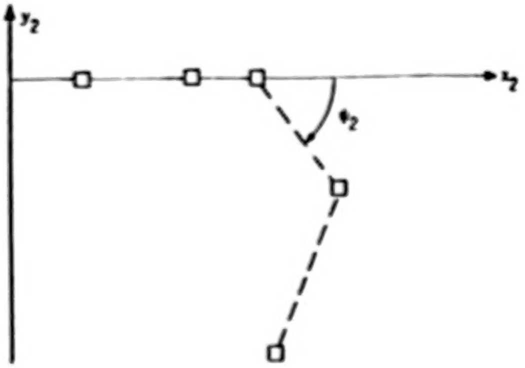
*Moretti, G., "Conformal Mappings for Computations of Steady, Three-Dimensional, Supersonic Flows," Numerical/Laboratory Computer Methods in Fluid Mechanics, ASME, 1976.

SEQUENCE OF TRANSFORMATIONS

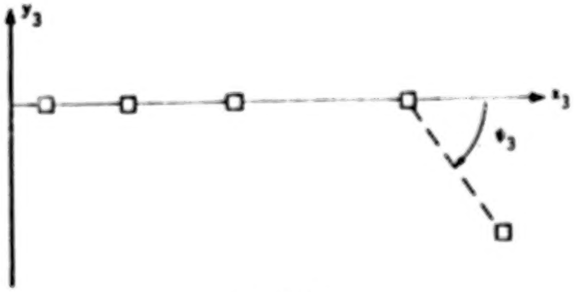
In the $j^{\underline{th}}$ mapping of the sequence, the transformation is centered around the hinge point $h_{j+1,j}$. The mappings have the property of keeping the hinge points, $h_{i,j}$ ($i \leq j+1$) on the horizontal axis, while mapping the hinge point $h_{j+2,j}$ onto the horizontal axis. Thus, after JA transformations, all JA+2 hinge points in the JA+1 space will lie on the horizontal axis. (Each mapping in this sequence may be considered a "point-wise Schwarz-Christoffel" transformation.) This figure illustrates the sequence of transformations for JA = 3.



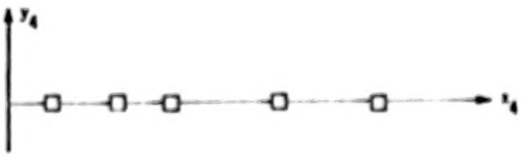
z_1 PLANE



z_2 PLANE



z_3 PLANE



z_4 PLANE

EXPONENTS OF TRANSFORMATIONS: $\delta_j = \frac{\pi}{\pi-\psi_j}$

TRANSFORMATIONS - CONTINUED

In order to establish a grid suitable for flow field calculations when the image of the body contour is a nearly horizontal surface, it is desirable to have the image of the centerline external to the body lie along the vertical axis. This is achieved using an additional conformal transformation, centered around the second hinge point, of the form

$$z_{JA+2} = (z_{JA+1} - h_{2,JA+1})^{1/2}.$$

The last transformation is a simple stretching (which is also conformal):

$$\zeta = \xi + i\eta = az_{JA+2} \quad .$$

(This stretching is used in the calculation procedure along the centerline.) This figure illustrates the body contour resulting in the ζ-plane for the case of a sphere with JA = 3, where the body surface is defined as $\eta = b(\xi)$.

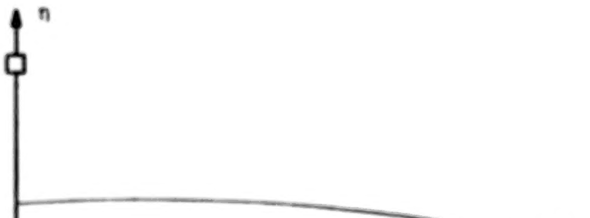
MAP CENTERLINE ONTO VERTICAL AXIS WITH

$$z_{JA+2} = (z_{JA+1} - h_{2,JA+1})^{1/2}$$

ALLOW FOR SIMPLE STRETCHING (REQUIRED FOR CENTERLINE TREATMENT) WITH

$$\zeta = \xi + i\eta = az_{JA+2}$$

RESULTING BODY CONTOUR:

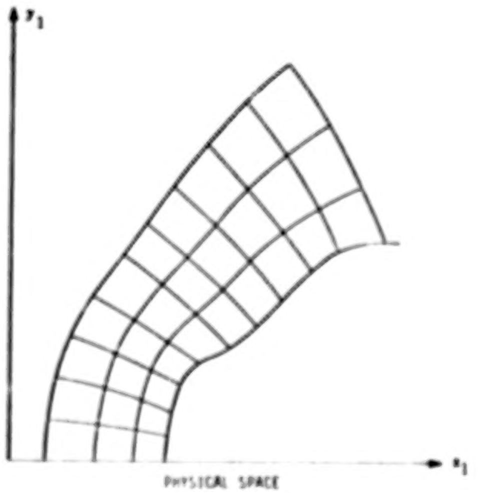


COMPUTATIONAL TRANSFORMATION

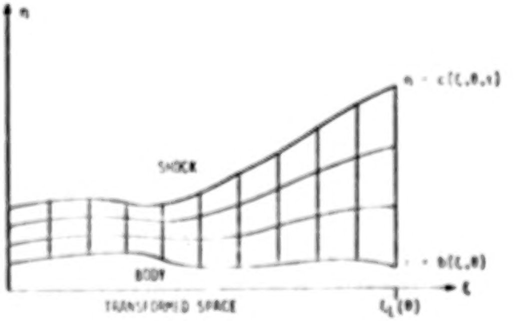
For the flow field calculation it is desirable to have equally spaced grid points. Thus, a transformation to a computational coordinate system (X,Y,Z) is used, in which grid points are equally spaced circumferentially in θ, longitudinally in ξ within each meridional plane, and in η between the body and the shock. It is important to note that the (X,Y,Z) system is not orthogonal, and that the computational transformation varies with time as the bow shock position varies during the time-dependent calculation. These sketches illustrate the computational grids resulting in a meridional plane in both physical ($z = x+iy$) and transformed ($\zeta = \xi+i\eta$) space for a typical ablated nosetip contour (with the shock layer thickness exaggerated for clarity).

DESIRE GRID POINTS EQUALLY SPACED IN ξ ALONG BODY, IN η BETWEEN BODY AND SHOCK, AND IN θ CIRCUMFERENTIALLY

$$X = \frac{\theta}{2\pi} \qquad Y = \frac{\xi}{\xi_L(\theta)} \qquad Z = \frac{\eta - b(\xi,\theta)}{c(\xi,\theta,\tau) - b(\xi,\theta)}$$



PHYSICAL SPACE



TRANSFORMED SPACE

PARAMETERS OF THE TRANSFORMATION

In transforming the governing equations from physical to the (X,Y,Z) computational coordinates, certain derivatives of the transformation are required. Because the transformation has been defined in analytic form, these derivatives can readily be evaluated analytically and are functions only of the hinge point locations. Within a meridional plane (ϕ = constant), the required derivatives are $g = \partial\zeta/\partial z$ and $\Phi = \partial(\log g)/\partial\zeta$. Circumferentially, the independent transformations in each meridional plane can be coupled to produce a three-dimensional transformation by assuming that hinge point locations can be expressed as $h_{i,j}(\phi)$. The required circumferential parameters of the transformation, ζ_ϕ and g_ϕ, can be evaluated analytically if each meridional plane has the same number of hinge points and assuming the form of interpolating functions for $h_{i,j}(\phi)$. Alternatively, it has been found to be sufficient to evaluate ζ_ϕ and g_ϕ from Taylor series expansions using data at computational (X,Y,Z) mesh points, with the forms of the resulting expressions shown in the figure.

REQUIRED IN WRITING GOVERNING EQUATIONS IN TRANSFORMED COORDINATES

$$g = \frac{\partial\zeta}{\partial z} = Ge^{i\omega} = \xi_x + i\eta_x = -i\xi_y + \eta_y$$

$$\Phi = \frac{\partial(\log g)}{\partial\zeta}$$

CAN BE EVALUATED ANALYTICALLY

CIRCUMFERENTIAL PARAMETERS OF THE TRANSFORMATION

ζ_ϕ, g_ϕ CAN BE EVALUATED ANALYTICALLY IF EACH MERIDIONAL PLANE HAS THE SAME NUMBER OF HINGE POINTS, ASSUMING INTERPOLATING FUNCTIONS FOR $h_{i,j}(\phi)$

ALTERNATIVELY, EVALUATE FROM TAYLOR SERIES EXPANSIONS:

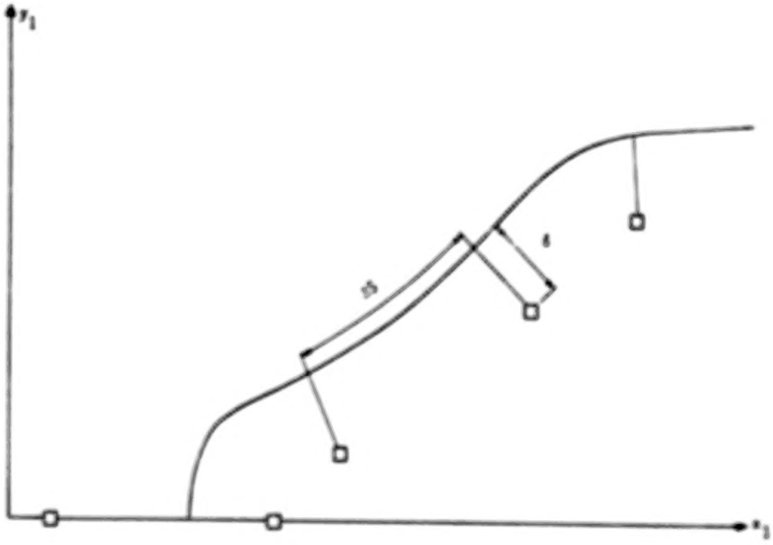
$$\zeta_\phi = \xi_\phi + i\eta_\phi = \frac{\zeta_2 - \zeta_1 - g(z_2 - z_1)}{\phi_2 - \phi_1}$$

$$g_\phi = \frac{g_2 - g_1 - g^2\,\Phi(z_2 - z_1)}{\phi_2 - \phi_1}$$

$(\ )_1 \rightarrow (X-\Delta X, Y, Z)$, $\quad (\ )_2 \rightarrow (X+\Delta X, Y, Z)$ IN COMPUTATIONAL MESH

AUTOMATIC GENERATION OF HINGE POINTS

To simplify the application of this coordinate transformation to the asymmetric nosetip flow field problem, the selection of hinge points that define the transformations has been automated. Within each meridional plane to be computed, body normals are constructed at points equally spaced in wetted length along the body profile. The hinge points are then selected to lie a distance δ inside the body along these normals. By relating δ to any convenient scale factor for a nosetip geometry, the only input required of the user of the code is the number of hinge points to be used. The locations of the first two hinge points (i.e., those that lie on the x axis) are the same in each meridional plane, in order to simplify the treatment of the centerline. Typically, no more than nine hinge points per meridional plane (JA = 7) are necessary for the nosetip flow field problem.



HINGE POINTS LOCATED DISTANCE δ ALONG INWARD BODY NORMALS, FROM BODY POINTS EQUALLY SPACED IN WETTED LENGTH

ONLY INPUT REQUIRED OF USER IS NUMBER OF HINGE POINTS TO BE USED IN EACH MERIDIONAL PLANE

TREATMENT OF CENTERLINE

The greatest complication encountered in the use of this 3-D coordinate transformation is the extra care that must be taken in treating the grid points on the centerline. Since the transformations in each meridional plane are independent, the scale factors $g = \partial\zeta/\partial z$ along the centerline will not be the same in each meridional plane. Thus, one computational grid point at the centerline will represent different physical points for each value of ϕ. To minimize these discrepancies, the stretching transformation $\zeta = az_{JA+2}$ is used to ensure that the images of the first hinge point are coincident in all meridional planes. The remaining discrepancies are small enough that simple linear interpolations can be used to account for differences in the scale factors.

In addition to the mapping complications along the centerline, the governing equations in cylindrical coordinates are singular along $y = 0$. This difficulty has been avoided by using a Cartesian (x_1, x_2, x_3) coordinate system for the centerline analysis. The required Cartesian derivatives can be expressed in terms of the radial derivative $\partial/\partial y$ in cylindrical coordinates for certain values of ϕ, as shown in this figure. The only restriction resulting from this analysis is that computational planes must be located at $\phi = 0$, $\pi/2$, π, and $3\pi/2$.

AT THE CENTERLINE $(y = 0)$, SCALE FACTORS $(g = \partial\zeta/\partial z)$ VARY WITH ϕ

STRETCHING TRANSFORMATION USED TO MINIMIZE DISCREPANCIES, WITH

$$a(\phi_k) = \frac{h_{1,JA+2}(\phi = 0)}{h_{1,JA+2}(\phi = \phi_k)}$$

CARTESIAN COORDINATES (x_1, x_2, x_3) USED IN CENTERLINE ANALYSIS

$$\frac{\partial}{\partial x_1} = \frac{\partial}{\partial x}$$

$$\frac{\partial}{\partial x_2} = \cos\phi \frac{\partial}{\partial y} - \frac{\sin\phi}{y} \frac{\partial}{\partial \phi}$$

$$\frac{\partial}{\partial x_3} = \sin\phi \frac{\partial}{\partial y} + \frac{\cos\phi}{y} \frac{\partial}{\partial \phi}$$

$$\lim_{y \to 0} \frac{1}{y} \frac{\partial}{\partial \phi} = \frac{\partial^2}{\partial y \partial \phi} \text{ FINITE,}$$

$$\frac{\partial}{\partial x_2} = \cos\phi \frac{\partial}{\partial y}, \quad \phi = 0, \pi$$

$$\frac{\partial}{\partial x_3} = \sin\phi \frac{\partial}{\partial y}, \quad \phi = \frac{\pi}{2}, \frac{3\pi}{2}$$

RESULTING FLOW FIELD CODE

The 3-D, time-dependent, inviscid nosetip flow field code that was developed using the 3-D coordinate transformation described here is called CM3DT (Conformal Mapping 3-D Transonic). This code can treat ideal or equilibrium real gas thermodynamics, has both pitch and yaw capability, and is able to treat weak embedded shocks on indented nosetips using the λ-differencing scheme*. To provide total body inviscid flow field capability, the CM3DT code has been coupled to the BMO/3IS**, NSWC/D3CSS^{+}, and STEIN^{++} afterbody codes. Complete details on the CM3DT analysis and results obtained with this code may be found in the following references:

Hall, D. W., "Inviscid Aerodynamic Predictions for Ballistic Reentry Vehicles with Ablated Nosetips," Ph.D. Dissertation University of Pennsylvania, 1979.

Hall, D. W., "Calculation of Inviscid Supersonic Flow over Ablated Nosetips," AIAA Paper 79-0342, January 1979.

CM3DT (CONFORMAL MAPPING 3-D TRANSONIC)
NOSETIP FLOW FIELD CODE

- IDEAL OR EQUILIBRIUM REAL GAS THERMODYNAMICS
- PITCH AND YAW CAPABILITY
- λ-DIFFERENCING SCHEME USED TO TREAT WEAK EMBEDDED SHOCKS ON INDENTED NOSETIPS
- COUPLED TO AFTERBODY CODES FOR TOTAL INVISCID FLOW FIELD CAPABILITY
  - BMO/3IS
  - NSWC/D3CSS
  - STEIN
- [illegible]

---

*Moretti, G., "An Old Integration Scheme for Compressible Flow Revisited, Refurbished, and Put to Work," Polytechnic Institute of New York, POLY-M/AE Report 78-22, September 1978.

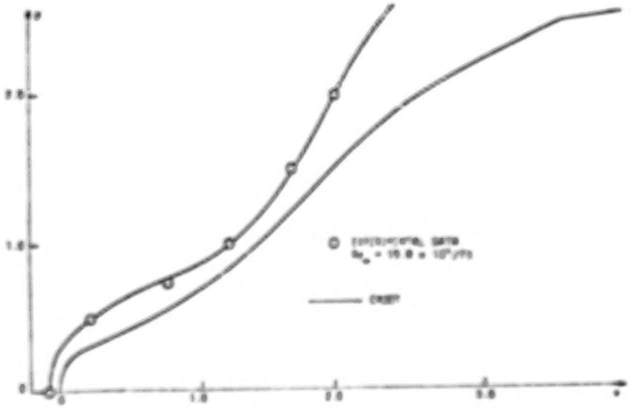
**Kyriss, C. L. and Harris, T. B., "A Three-Dimensional Flow Field Computer Program for Maneuvering and Ballistic Reentry Vehicles," 10th U.S. Navy Symposium on Aeroballistics, July 1975; also, Daywitt, J., Brant, D., and Bosworth, F., "Computational Technique for Three-Dimensional Inviscid Flow Fields about Reentry Vehicles, Volume I: Numerical Analysis," SAMSO TR-79-5, April 1978.

$^{+}$Solomon, J. M., Ciment, M., Ferguson, R. E., Bell, J. B., and Wardlaw, A. B., Jr., "A Program for Computing Steady Inviscid Three-Dimensional Supersonic Flow on Reentry Vehicles, Volume I: Analysis and Programming," Naval Surface Weapons Center, NSWC/WOL/TR 77-28, February 1977.

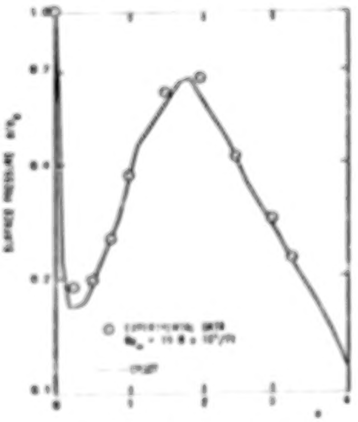
$^{++}$Marconi, F., Salas, M., and Yaeger, L., "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow around Real Configurations, Volume I. Computational Technique," NASA CR-2675, April 1976.

CM3DT RESULTS

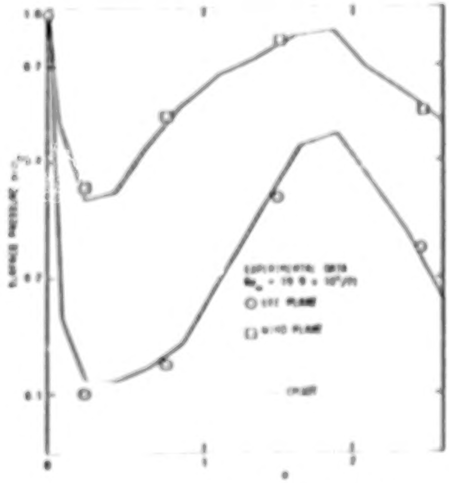
This figure presents some typical results obtained with the CM3DT inviscid nosetip flow field code. Shown are comparisons of predictions to data obtained for the PANT Triconic shape* at $M_\infty = 5$. It is significant that attempts to compute the flow over this slender shape using a time-dependent code formulated in a spherical coordinate system were unsuccessful. CM3DT, with its body-oriented coordinate system, was able to obtain converged solutions for this shape, with the predictions agreeing well with the data, as seen in this figure.



CM3DT Shock Shape Prediction for the PANT Triconic at $M_\infty = 5$, $\alpha = 0°$



Surface Pressure Predictions for the PANT Triconic at $M_\infty = 5$, $\alpha = 0°$



Surface Pressure Predictions for the PANT Triconic at $M_\infty = 5$, $\alpha = 10°$

(Note: 1 ft = 0.3048 m)

---

*Abbett, M. J. and Davis, J. E., "Interim Report, Passive Nosetip Technology (PANT) Program, Volume IV. Heat Transfer and Pressure Distribution on Ablated Shapes, Part II. Data Correlation and Analysis," Space and Missile Systems Organization, TR-74-86, January 1974.

CM3DT - RUN TIMES

On a CDC Cyber 176 computer, the CM3DT inviscid nosetip code with λ-differencing requires approximately 0.00045 CP seconds per grid point per time step (iteration). Typically, 400-500 time steps are required to obtain a converged solution. It is estimated that the computer time required for a solution has been increased by approximately 20% by using the 3-D coordinate transformation described here, when the parameters of the transformation on the moving grid are updated every ten time steps. When compared to the standard MacCormack differencing scheme, the use of λ-differencing scheme increases the run time requirements approximately 50% for this code.

ON A CDC CYBER 176, CM3DT REQUIRES 0.00045 CP SECS/POINT/STEP

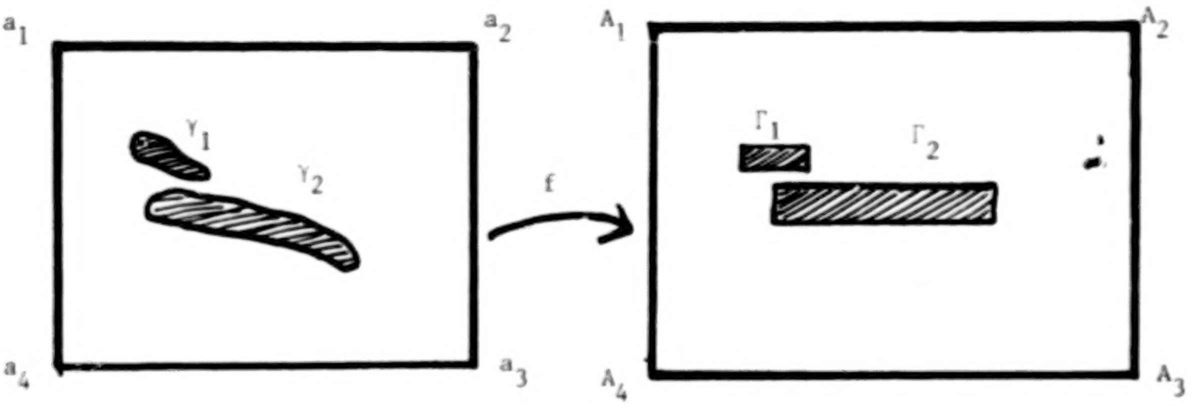
FOR IDEAL GAS CALCULATIONS WITH λ-DIFFERENCING

- 20% PENALTY INCURRED FOR COORDINATE TRANSFORMATION (PARAMETERS ON MOVING GRID UPDATED EVERY 10 TIME STEPS)

- 50% PENALTY INCURRED FOR λ-DIFFERENCING (RELATIVE TO MAC CORMACK DIFFERENCING)

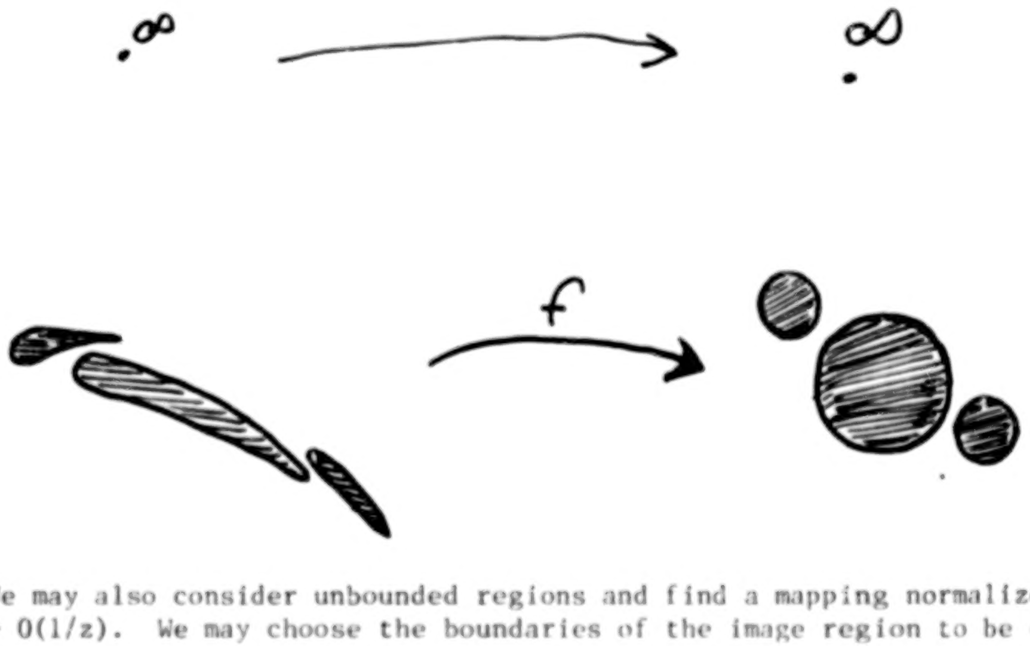
CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES

Andrew N. Harrington
School of Mathematics
Georgia Institute of Technology



f conformal $\quad f(a_j) = A_j,\ \gamma_j \xrightarrow{f} \Gamma_j$

The author has developed and implemented a numerical procedure to compute the conformal mapping of a given n-tuply connected region onto a region with any specified boundary shapes and with several possible normalizations. If we start with a region whose outer boundary is a rectangle, we may arrange that the outer boundary of the image region is also a rectangle and the vertices map to vertices. We may choose the inner boundaries to map to rectangles or to any other shapes.



We may also consider unbounded regions and find a mapping normalized at ∞ $z + O(1/z)$. We may choose the boundaries of the image region to be circles or any other shapes.

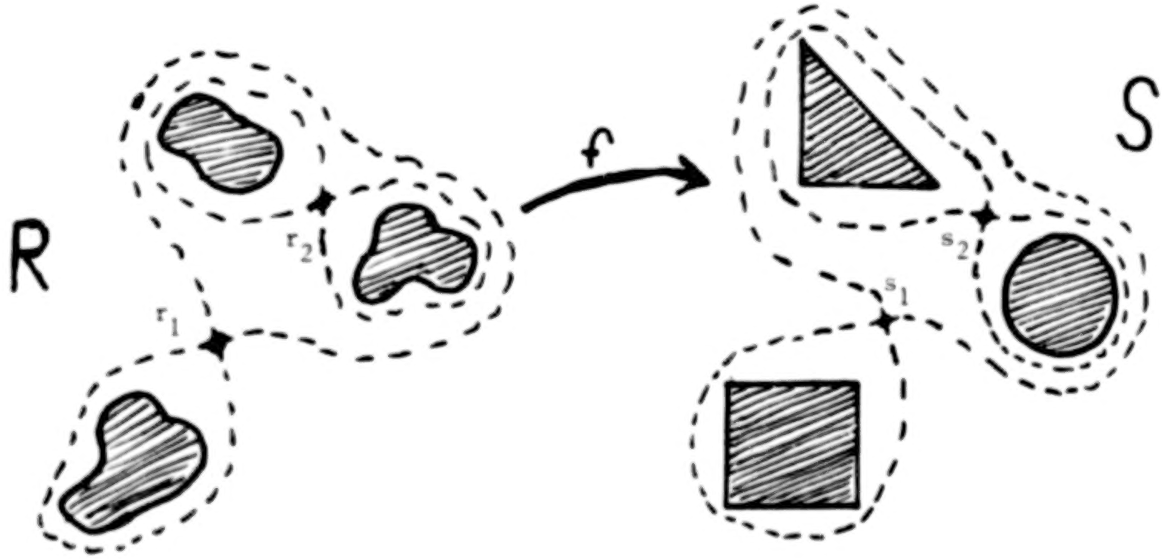
Method

Though we may specify boundary shapes and orientations arbitrarily, the proper translation and magnification parameters must be calculated to determine the image domain and the mapping. For example, in order to find a conformal mapping between n-tuply connected regions R and S containing ∞ with $f(\infty) = \infty$, we must satisfy conditions on G_R and G_S, the analytic completions of the Green's functions for R and S with pole at ∞. We must have

$$G_R(r_j) = G_S(s_j) \qquad j = 1, 2, \ldots n-1$$

where r_j and s_j, $j = 1, 2, \ldots n-1$, are the critical points for G_S and G_R labeled in the figure. Using Symm's method to approximate Green's functions one may easily calculate the appropriate parameters. Then $G_S(f(z)) = G_R(z)$.

The dotted curves are the level curves of Re G_R and Re G_S which branch at the critical points.



BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS

Edward J. Kowalski
Boeing Military Airplane Company
Advanced Airplane Branch
Seattle, Washington 98124

A computational region of arbitrary cross section presents a significant problem in the generation of a mesh. Simple orthogonal meshes are difficult to use because the mesh points do not naturally fall on the region's boundaries. Differencing and interpolation schemes become complex and cumbersome, and it is difficult to extend these schemes to higher order because of the complex logic required. Higher order schemes are desirable as they allow calculation of a flow to a given level of accuracy with a lower mesh density and hence less storage than a lower order scheme. High accuracy solutions are possible for a region of arbitrary cross section when a boundary-fitted computational mesh is employed. A boundary-fitted mesh is defined as a mesh in which the boundary (i.e., a duct wall) is coincident with the mesh points that are used for finite difference expressions at, and adjacent to, the boundary. Interpolation is not required, and extension to higher order differencing is straightforward. This is a significant benefit when the boundary conditions have a dominant influence on the solution.

This paper will discuss the application of Smith and Wiegel's method for generating boundary fitted coordinate systems (discussed in their AIAA-80-0192 paper entitled, "Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Simulation") for two practical flow problems characterized by complex surface geometry:

- radial mixer lobe
- subsonic inlet designed for high angle-of-attack capability

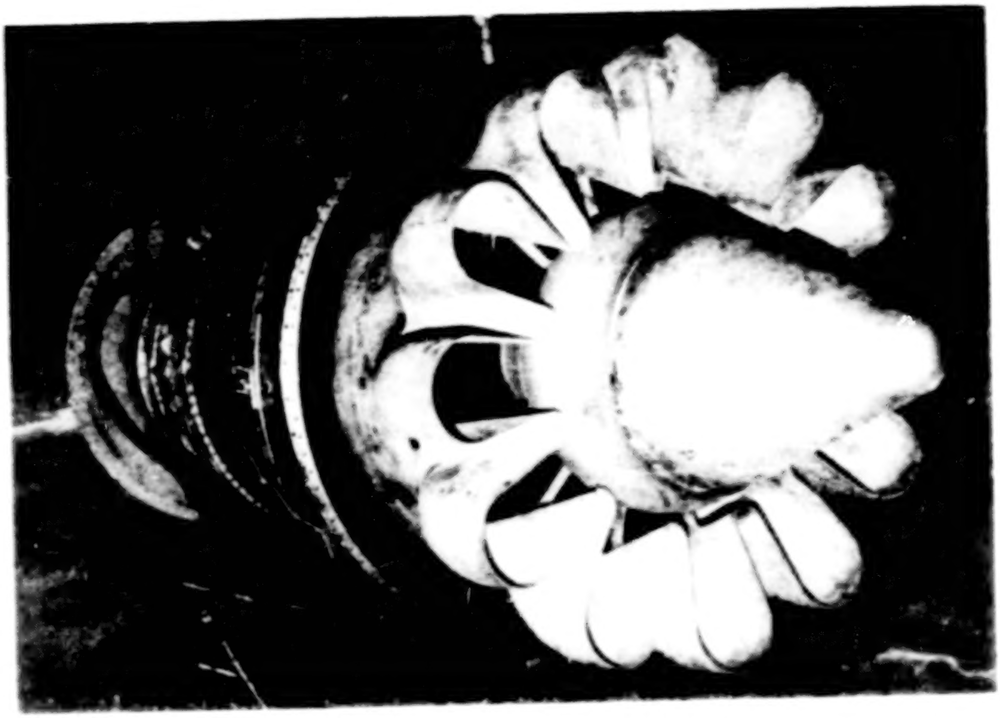
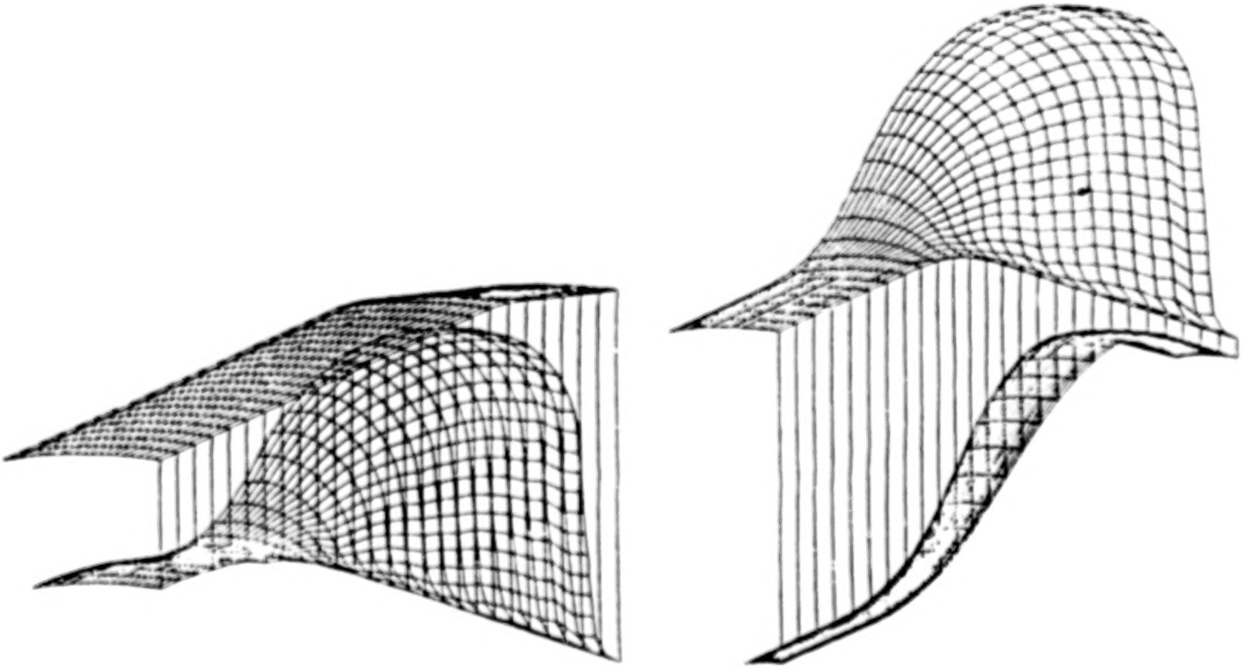


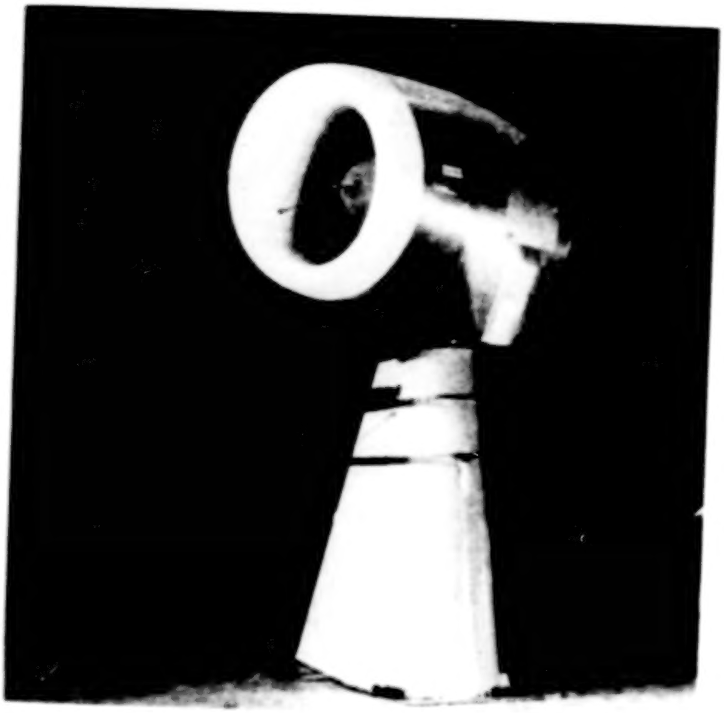
Figure 1.- Full scale forced mixer.



Secondary stream

Primary stream

Figure 2.- Radial mixer lobe.



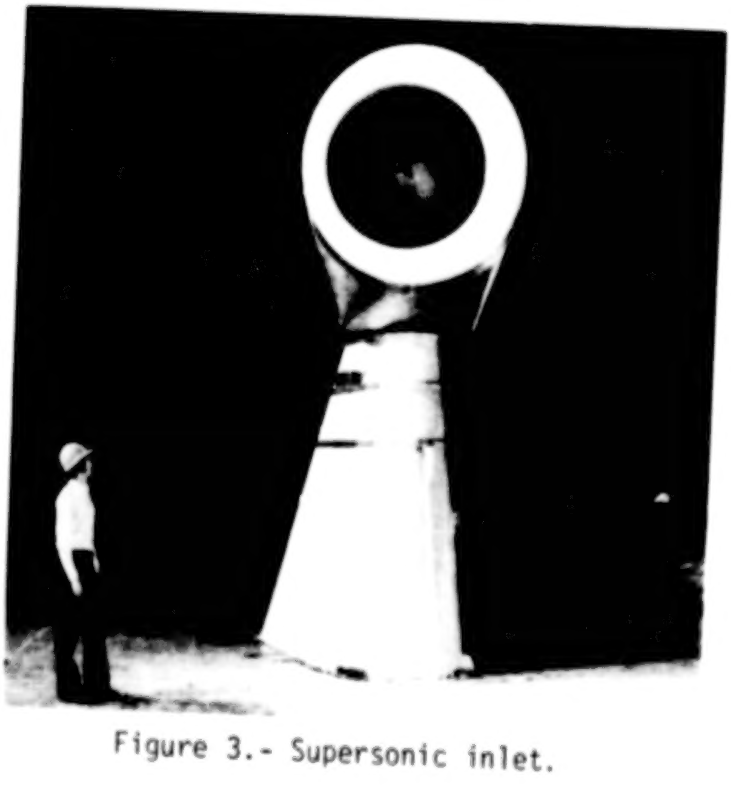


Figure 3.- Supersonic inlet.

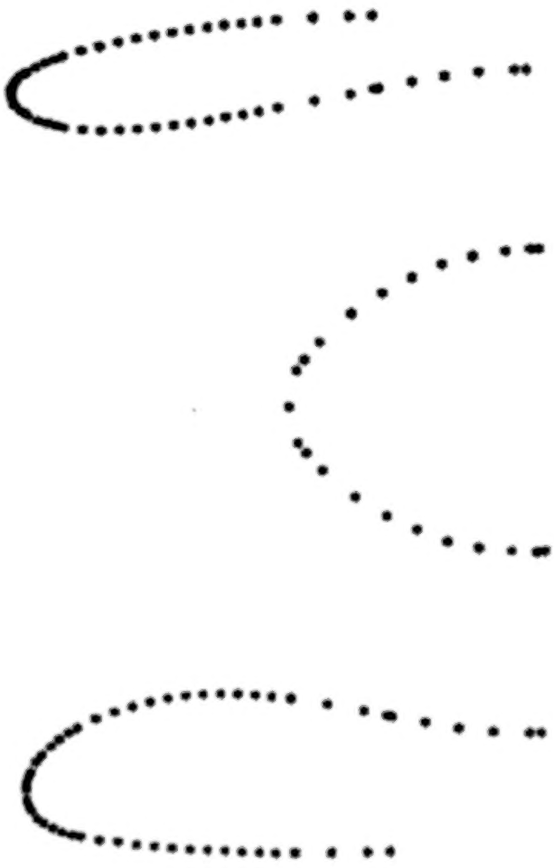


Figure 4.- Inlet contour.

In the method of Smith and Wiegel, two disconnected boundaries are defined and an explicit functional relation is used to establish the transformation between the physical domain and the computational domain.

The physical domain is defined by a cartesian coordinate system; the computational domain is defined with the variables ξ , η and ζ with the values:

$$0 \leq \xi \leq 1$$

$$0 \leq \eta \leq 1$$

$$0 \leq \zeta \leq 1$$

Two possible connecting functions are suggested: linear and a cubic parametric polynomial. The following cubic polynomial equation was used to generate meshes for both the lobe mixer and the subsonic inlet:

$$\begin{aligned}
x &= X_1(\xi,\zeta)f_1(\eta) + X_2(\xi,\zeta)f_2(\eta) + \frac{dX_1}{d\eta}(\xi,\zeta)f_3(\eta) \\
&\quad + \frac{dX_2}{d\eta}(\xi,\zeta)f_4(\eta) \\
y &= Y_1(\xi,\zeta)f_1(\eta) + Y_2(\xi,\zeta)f_2(\eta) + \frac{dY_1}{d\eta}(\xi,\zeta)f_3(\eta) \\
&\quad + \frac{dY_2}{d\eta}(\xi,\zeta)f_4(\eta) \\
z &= Z_1(\xi,\zeta)f_1(\eta) + Z_2(\xi,\zeta)f_2(\eta) + \frac{dZ_1}{d\eta}(\xi,\zeta)f_3(\eta) \\
&\quad + \frac{dZ_2}{d\eta}(\xi,\zeta)f_4(\eta)
\end{aligned}$$

where:

$X_\ell(\xi,\zeta), Y_\ell(\xi,\zeta), Z_\ell(\xi,\zeta)$, $\ell = 1,2$ are the boundary points in the physical domain

$\frac{dX_\ell}{d\eta}(\xi,\zeta)$, $\frac{dY_\ell}{d\eta}(\xi,\zeta)$ $\frac{dZ_\ell}{d\eta}(\xi,\zeta)$, $\ell = 1,2$ are the derivatives of the boundary points in the physical domain

$$f_1(\eta) = 2\eta^3 - 3\eta^2 + 1$$
$$f_2(\eta) = -2\eta^3 + 3\eta^2$$
$$f_3(\eta) = \eta^3 - 2\eta^2 + \eta$$
$$f_4(\eta) = \eta^3 - \eta^2$$

The cubic connecting function forces orthogonality at the boundaries of the physical domain by calculating the derivatives $\frac{dX_\ell}{d\eta}(\xi,\zeta)$ $\frac{dY_\ell}{d\eta}(\xi,\zeta)$ and $\frac{dZ_\ell}{d\eta}(\xi,\zeta)$ from the cross product of the tangential derivatives and then dividing by the magnitude of the normal vector.

Four extensions of the Smith and Wiegel method were necessary in order to successfully apply their technique to the mixer lobe and subsonic inlet.

First, because of the nature of the mixer and inlet geometries, points defining the boundaries had to be positioned using a geometric progression.

$$S = a + ar + ar^2 + \ldots + ar^{N-1}$$
$$= \frac{a(1-r^N)}{1-r}$$

where

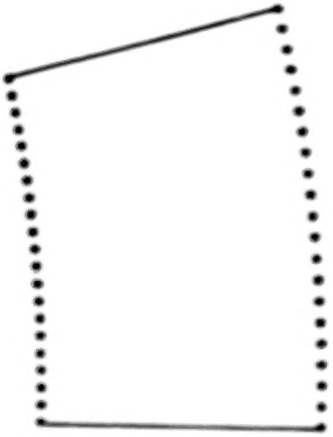
S = the total length of the boundary

a = first increment

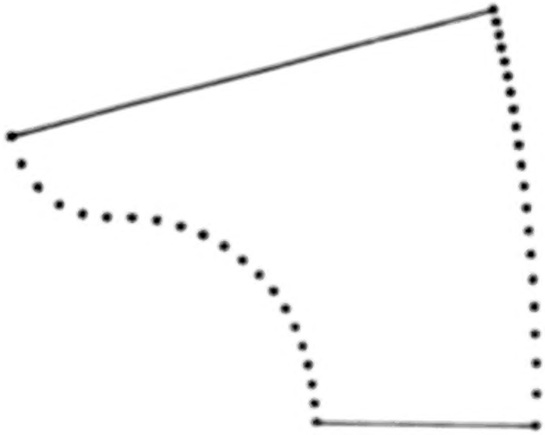
r = scale factor

N = number of cells (one less the number of boundary points)

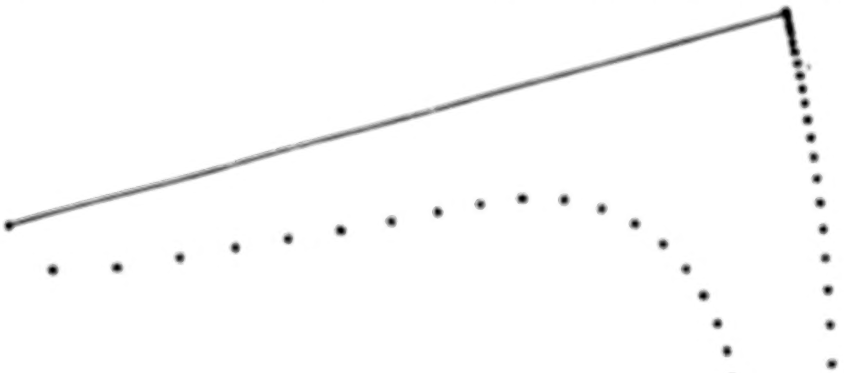
For the mixer, the scale factor r was varied linearly from r = 1 at the mixer entrance plane (where the boundary is an arc) to $r = r_{max}$ at the mixer exit plane (where the boundary is highly distorted). This makes it possible to force the mesh to migrate to regions of interest without causing significant distortions in the mesh from plane to plane. The optimal distribution of mesh occurred when the upper and lower boundary mesh points were stretched in opposite directions.



SECONDARY LOBE - PLANE 94
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000
SLOPE SCALER LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000

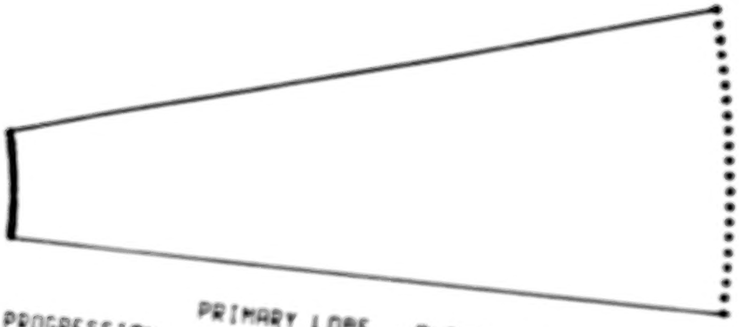


SECONDARY LOBE - PLANE 107
GEOMETRIC PROGRESSION LOWER BOUNDARY 0.9750 UPPER BOUNDARY 1.0500
SLOPE SCALER LOWER BOUNDARY 1.0250 UPPER BOUNDARY 1.0500



SECONDARY LOBE - PLANE 120
GEOMETRIC PROGRESSION LOWER BOUNDARY 0.9500 UPPER BOUNDARY 1.1000
SLOPE SCALER LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.1000

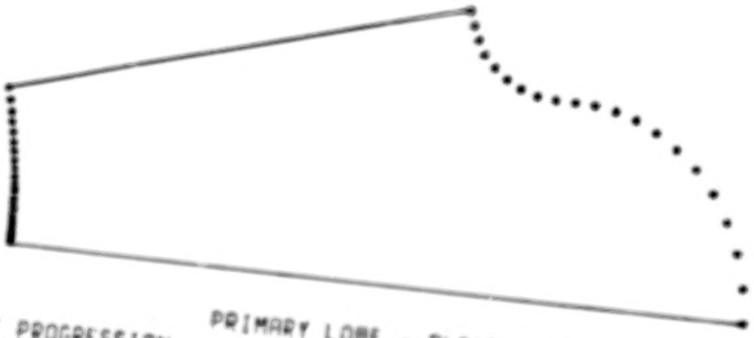
Figure 5.- Geometric progression for boundary points for secondary stream.



PRIMARY LOBE - PLANE 94

GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000

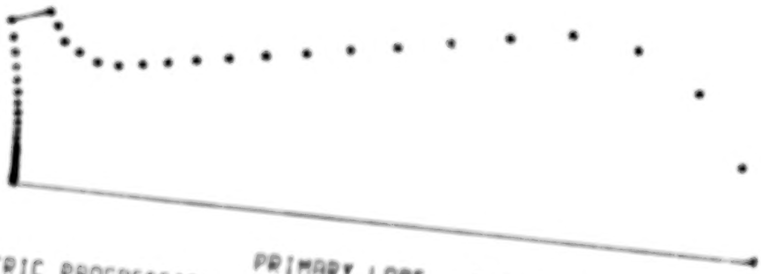
SLOPE SCALER LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000



PRIMARY LOBE - PLANE 107

GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.0500

SLOPE SCALER LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.0500



PRIMARY LOBE - PLANE 120

GEOMETRIC PROGRESSION LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000

SLOPE SCALER LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000

Figure 6.- Geometric progression of boundary points for primary stream.

The inlet has certain regions (hilite, throat, etc.) which require a fine computational mesh to insure a detailed analysis. For this reason, four regions along each inlet contour and five regions along the boundary of the analysis domain required individual geometric progressions. The scale factor, r, and the number of cells, N, of each region must be chosen to insure a smooth progression in cell length along each of the boundaries.

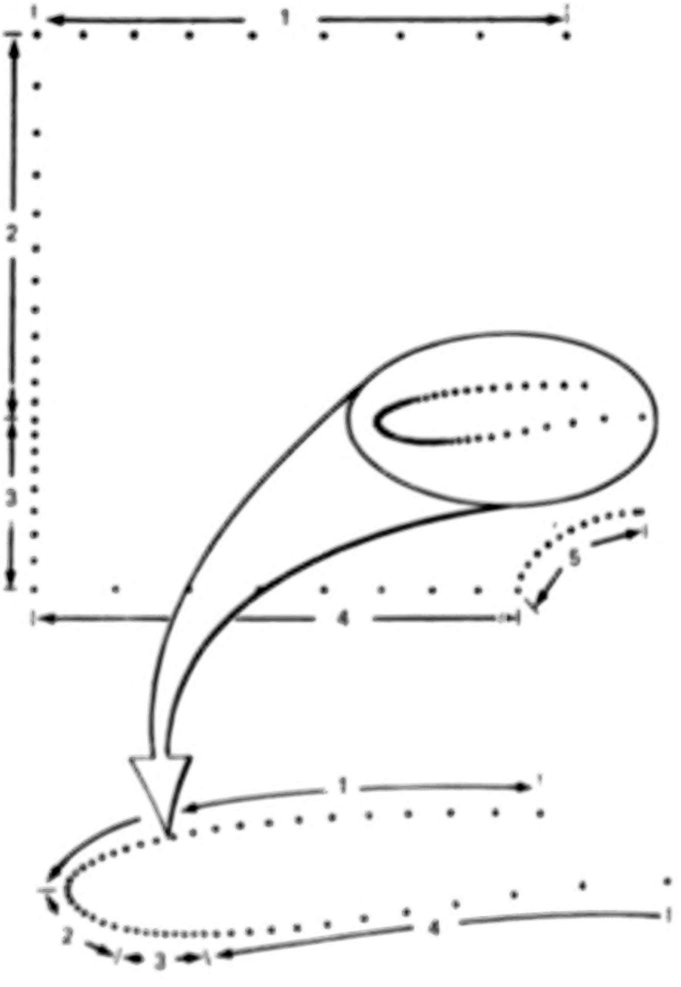
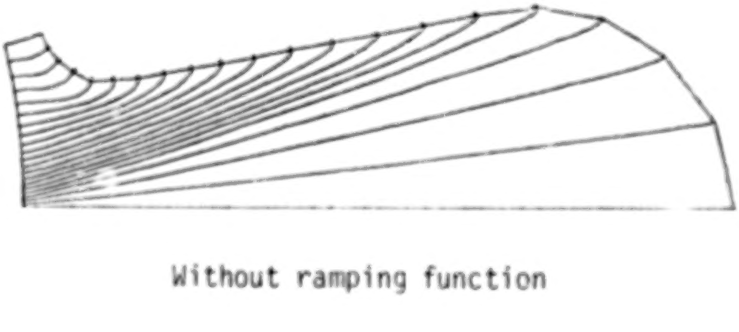


Figure 7.- Geometric progression regions along inlet contour and analysis boundary.

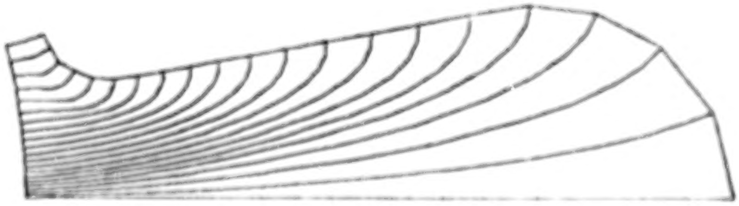
The second extension uses a ramping function to regulate the dependence of the connecting function on the boundary slope. This connecting function is an explicit functional relation used to establish the transformation between the physical domain and the computational domain.

For the mixer lobe, this dependence was regulated to redistribute the internal mesh points and reduce mesh skewness.

In the case of the subsonic inlet, it was found that a constant value for each plane was sufficient to insure against mesh line cross-over.

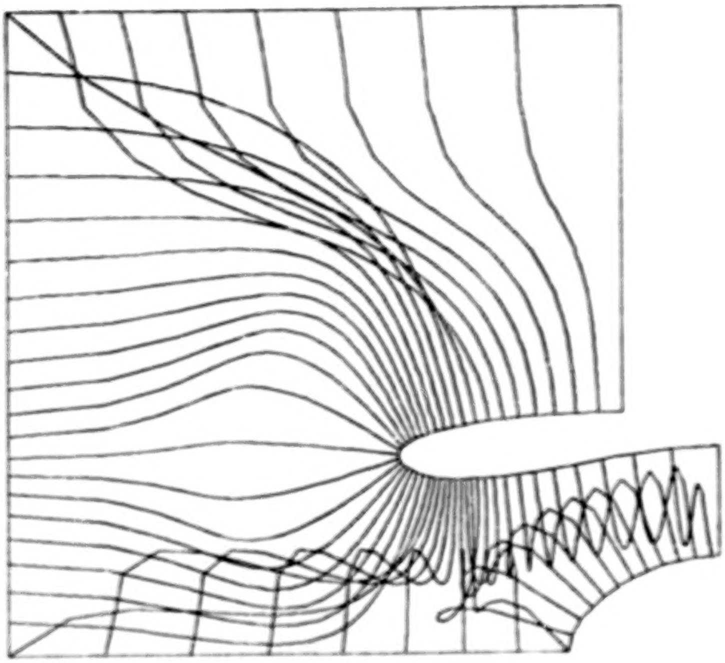


Without ramping function

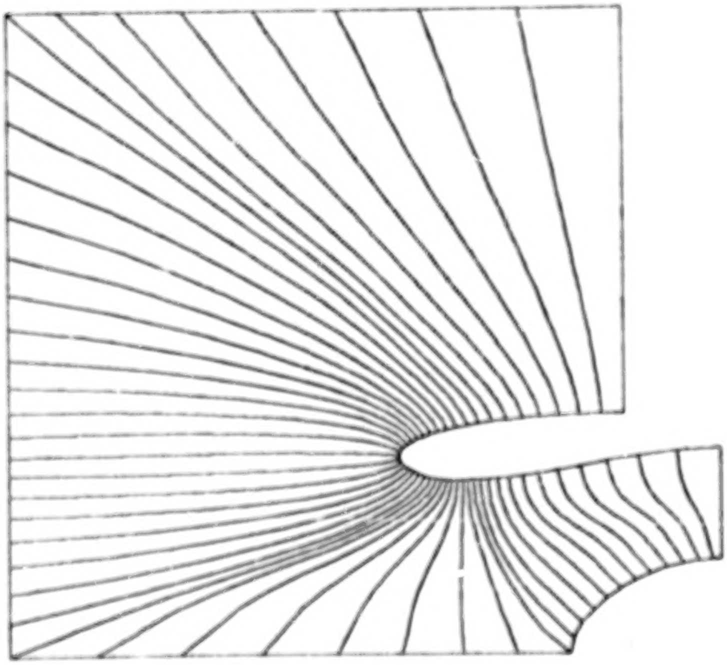


With ramping function

Figure 8.- Connecting function dependency on boundary slope.



Without ramping function

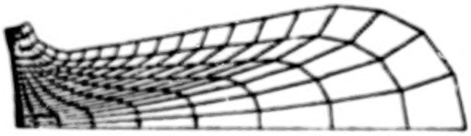


With ramping function

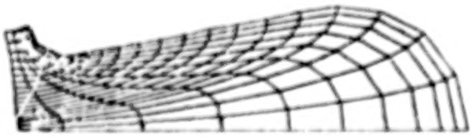
Figure 9.- Connecting function dependency on boundary slope.

The third extension utilizes the concentration function suggested by Smith and Wiegel, but uses it to force the mesh in the direction of both boundaries of the mixer lobe. More mesh was then needed to be linearly added to fill the void created by this mesh concentration.

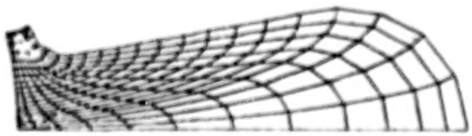
The inlet only required the mesh to be forced towards the inlet contour. A concentrated mesh was assumed unnecessary along the spinner boundary; it was felt that for a potential flow analysis the flow about the spinner would not propagate upstream and affect the solution at the regions of interest (hilite, throat, etc.). The mesh concentration for both the mixer and the inlet permits flow analysis within the boundary layers.



Mesh concentrated towards
inner boundary



Mesh concentrated towards
outer boundary



Mesh concentrated towards
both boundaries

Figure 10.- Mesh concentration.

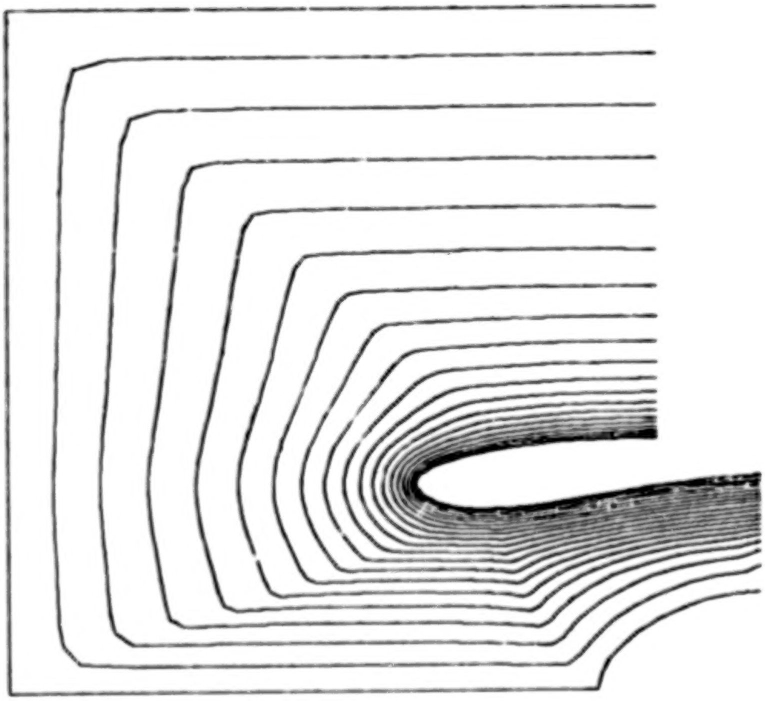


Figure 11.- Mesh concentration.

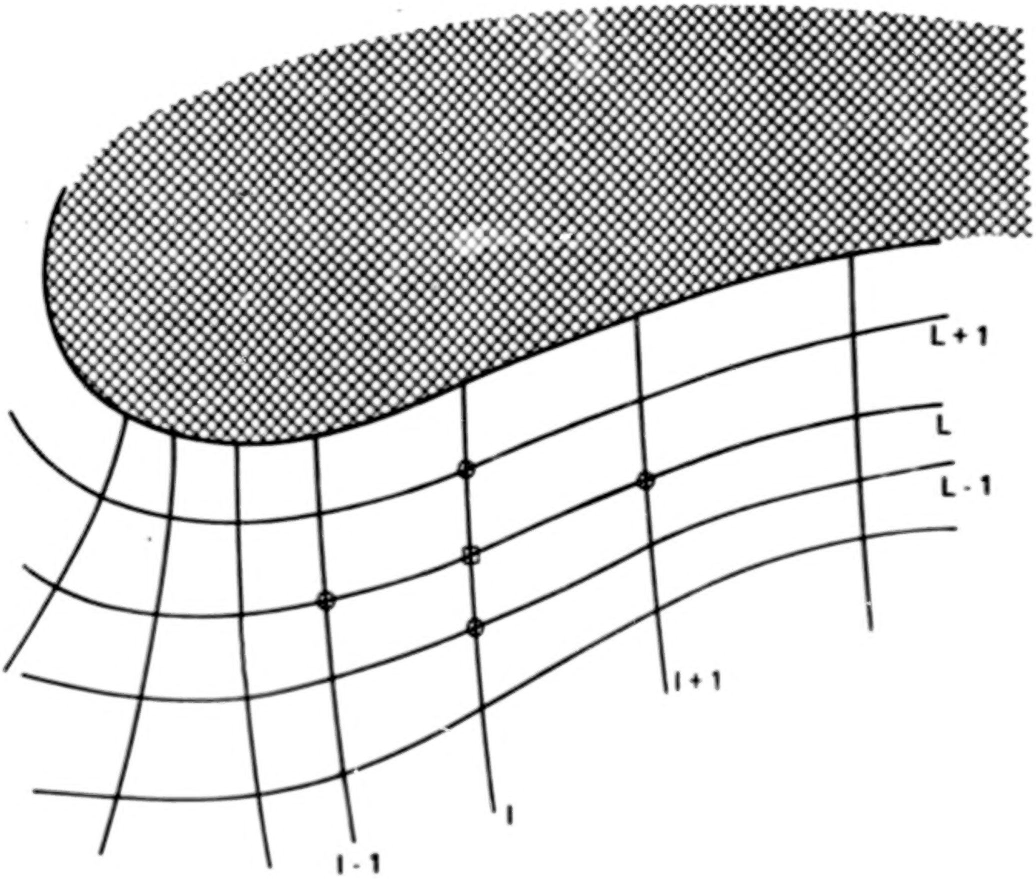
The fourth extension applies to the subsonic inlet only. It was necessary to produce a computational mesh which possessed a smooth progression of cell metrics and cell volumes in all directions to allow a solution process of a flow analyser to use the grid efficiently. The interior points of the computational mesh were "smoothed" by a multiple application of a five point diffusion operator:

$$X(L,I)_{new} = \alpha \left\{ X(L-1,I) + X(L+1,I) + X(L,I-1) + X(L,I+1) - 4 * X(L,I)_{old} \right\}$$

$$Y(L,I)_{new} = \alpha \left\{ Y(L-1,I) + Y(L+1,I) + Y(L,I+1) + Y(L,I+1) - 4 \bullet Y(L,I)_{old} \right\}$$

The value of α and the number of times of application were determined by trial and error.

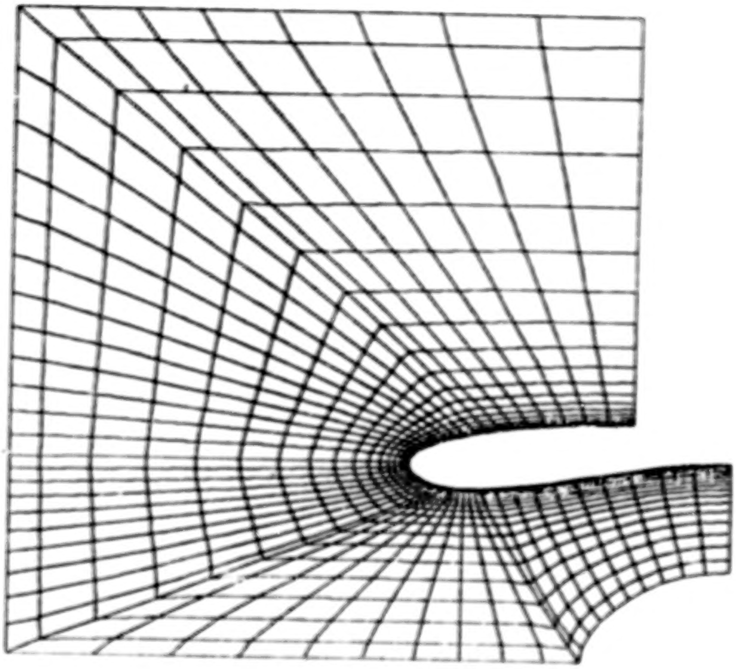
The "smoothed" boundary points could not be determined from the five point diffusion operator since one of the required smoothing points would be outside the mesh region. Their values were determined from the intersection of the lines defined by the "smoothed" interior mesh points and the boundaries.



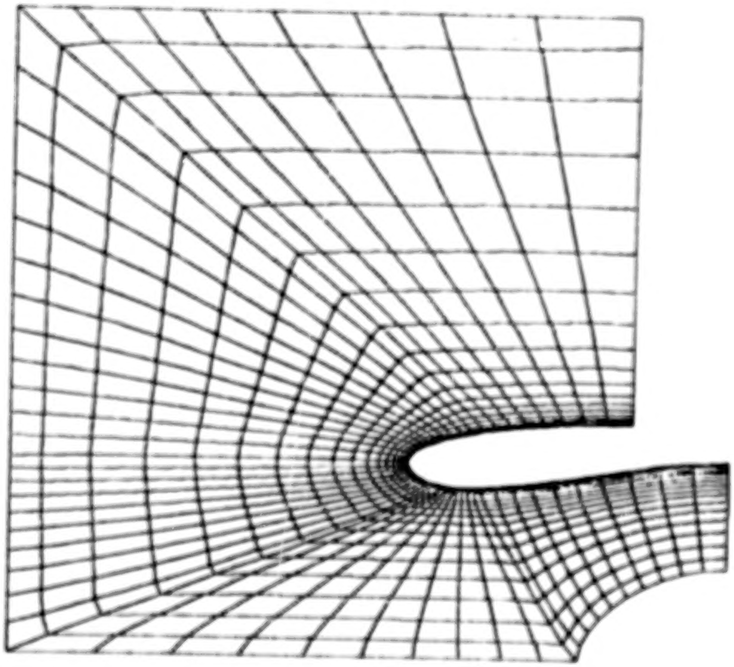
$$X(L,I)_{new} = \alpha \left\{ X(L-1,I) + X(L+1,I) + X(L,I-1) + X(L,I+1) - 4 * X(L,I)_{old} \right\}$$

$$Y(L,I)_{new} = \alpha \left\{ Y(L-1,I) + Y(L+1,I) + Y(L,I+1) + Y(L,I+1) - 4 \cdot Y(L,I)_{old} \right\}$$

Figure 12.- Five point diffusion operator.

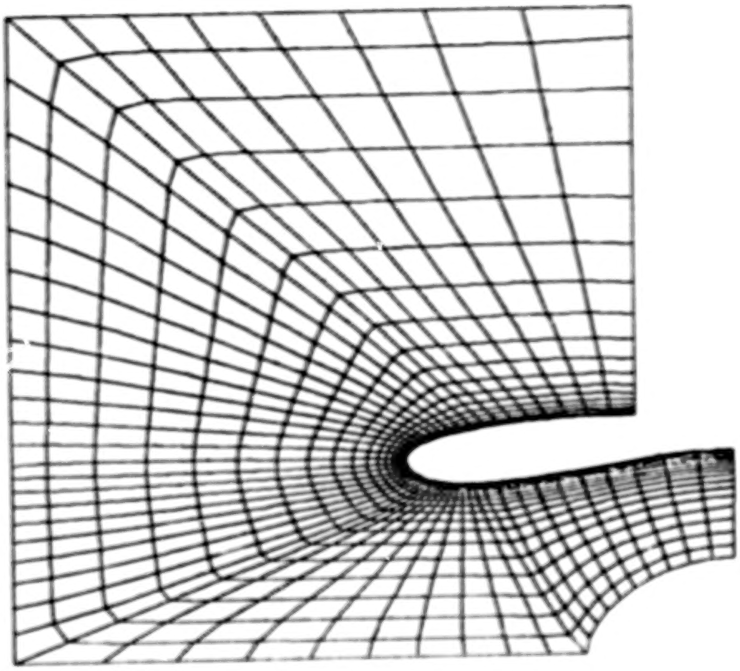


α = 0.05; Number of interations = 1

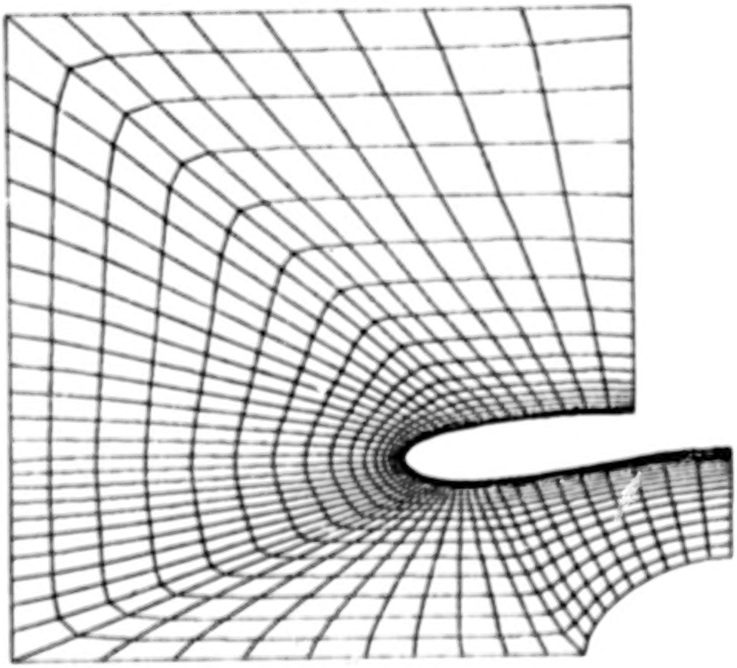


α = 0.05; Number of interations = 3

Figure 13.- "Smoothed" computational mesh.

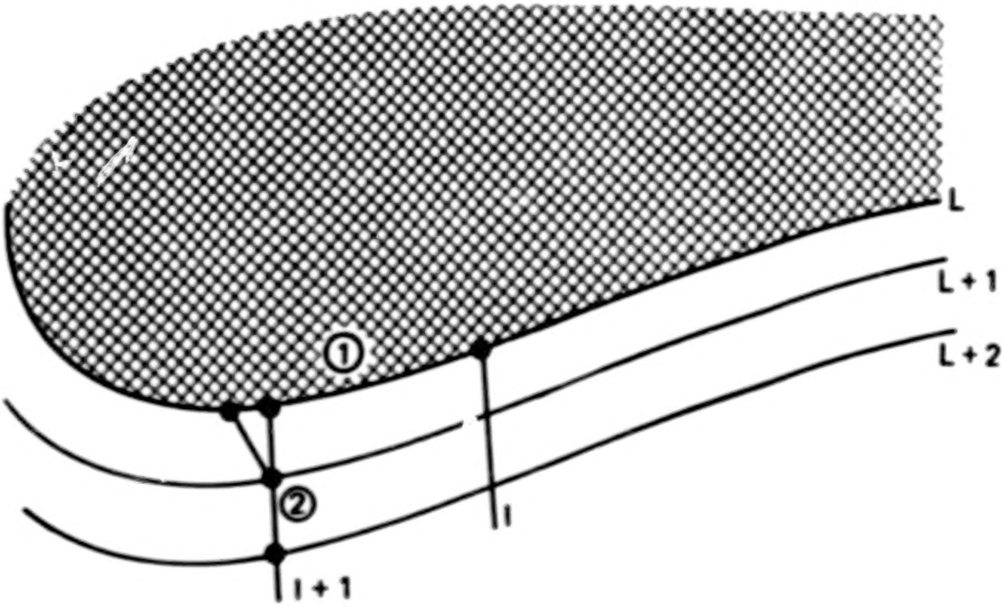


α = 0.05; Number of interations = 6



α = 0.05; Number of interations = 12

Figure 13.- Concluded.



slope of segment ①

$$M_1 = \frac{Y(L, I) - Y(L, I+1)}{X(L, I) - X(L, I+1)}$$

equation of segment ①

$$Y - Y(L,I) = M_1 \left\{ X - X(L,I) \right\} \qquad \text{②}$$

slope segment ②

$$M_2 = \frac{Y(L+1, I+1) - Y(L+2, I+1)}{X(L+1, I+1) - X(L+2, I+1)}$$

equation of segment ②

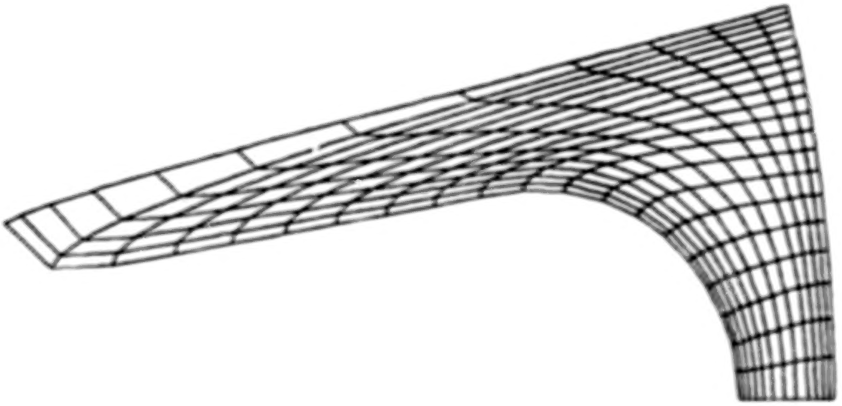
$$Y - Y(L+1, I+1) = M_2 \left\{ X - X(L+1, I+1) \right\} \qquad \text{①}$$

since a line thru segment ② intersects segment ①,
the X's and Y's of equations ① & ② equal each other.
Solving for X:

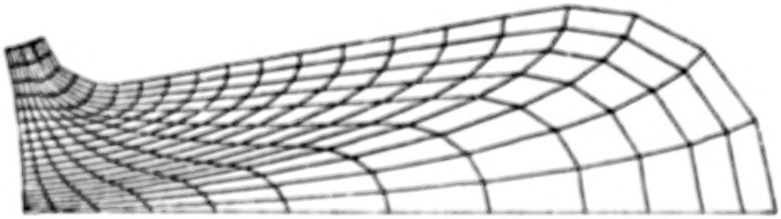
$$X(L, I+1)_{new} = \frac{M_1 \left\{ X(L, I) \right\} - M_2 \left\{ X(L+1, I+1) \right\} + Y(L+1, I+1) - Y(L, I)}{M_1 - M_2}$$

Solving for Y:

$$Y(L, I+1)_{new} = M_1 \left\{ X(L, I+1)_{new} - X(L, I) \right\} + Y(L, I)$$



SECONDARY LOBE - PLANE 120
GEOMETRIC PROGRESSION LOWER BOUNDARY 0.9500 UPPER BOUNDARY 1.1000
SLOPE SCALER LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.1000



PRIMARY LOBE - PLANE 120
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000
SLOPE SCALER LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000

Figure 14.- Example mesh for last mixer plane.

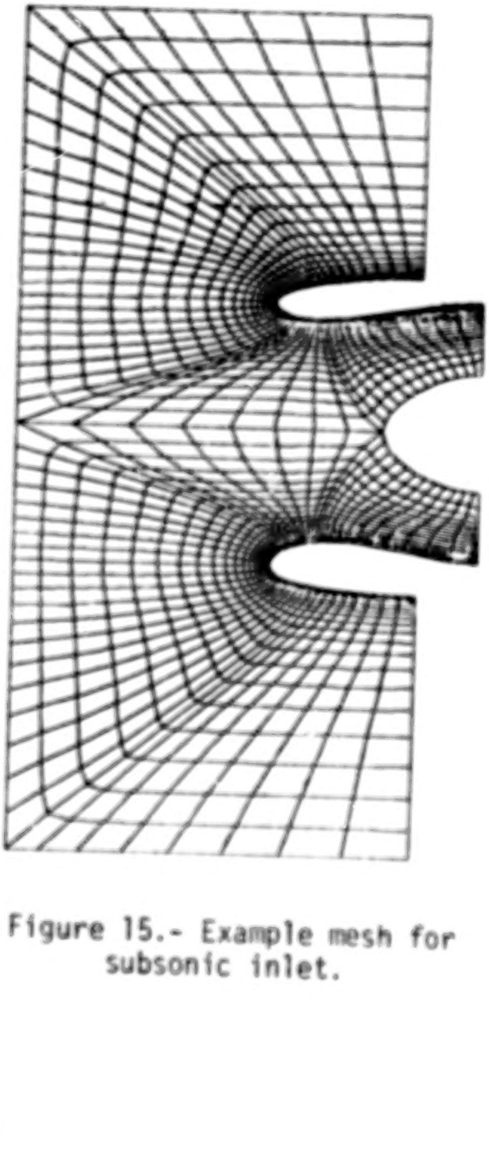


Figure 15.- Example mesh for subsonic inlet.

Conclusions

The method of Smith and Wiegel can be used to generate meshes for mixer lobes and subsonic inlets that are compatible with flow analysis codes requiring a boundary fitted coordinate system. Successful application of this mesh generator required development of procedures to distribute the mesh points along the boundaries, to regulate the dependence of the connecting function to the local boundary slope, to concentrate the mesh into regions of special interest, and to modify the mesh grid so that it possessed a smooth progression of cell metrics and cell volumes in all directions. The method of Smith and Wiegel when coupled with the extensions mentioned above has proven to be easy to use and control for the inlet and mixer lobe geometries investigated.

The next step is the formulation of a truncation error monitor for arbitrary meshes. This monitor will define where in an analysis domain the grid length scales must be changed and by what amount in order to equalize truncation errors over the entire analysis domain. Once these errors have been equalized, this same monitor will use several levels of grid distribution (of the above analysis grid) to then make estimates of the absolute truncation error spectrum. This work is currently under contract with the NASA Langley Research Center.

BLANK PAGE

BLANK PAGE

Grid Generation for General Three-Dimensional Configurations

K. D. Lee, M. Huang, N. J. Yu and P. E. Rubbert
The Boeing Company
Seattle, Washington

Abstract

The objective of the present study is to construct a suitable grid system for complex 3-D configurations such as a wing/body/nacelle shape for the solution of nonlinear transonic flow problems. Two approaches have been explored based on Thompson's body-fitted coordinate concept. The most general approach is to divide the computational domain into multiple rectangular blocks where the configuration itself is also represented by a set of blocks, whose structure follows the natural lines of the configuration. The block-structured grid system is adaptable to complex configurations and gives good grid quality near physical corners. However, it introduces algorithm issues for the flow solution concerning the treatment of nonanalytic grid block boundaries and nonstandard grid cells. These issues have been explored in relation to the grid generation. A more limited approach treats a wing/body configuration with only a single rectangular block in computational space. In this treatment the issues involving nonstandard cells are avoided, but other limitations on grid resolution appear. Both a linear and a nonlinear system of grid generation equations have been developed including methods of grid control. The linear method can generate grids of comparable quality with order-of-magnitude less cost. Its disadvantage is the greater possibility of ill-conditioned grids which, however, can be easily controlled in the block-structured grid system.

Grid Generation Equations

1 Linear System

$$\vec{x}_{\xi\xi} + B\vec{x}_{\eta\eta} + C\vec{x}_{\zeta\zeta} + D\vec{x}_{\xi} + E\vec{x}_{\eta} + F\vec{x}_{\zeta} + G = 0$$

$\vec{x} = (x, y, z)$

B to G: grid control
functions of ξ, η, and/or ζ

2 Nonlinear System

$$A\left(\vec{x}_{\xi\xi} + \frac{P}{J^2A}\vec{x}_{\xi}\right) + B\left(\vec{x}_{\eta\eta} + \frac{Q}{J^2B}\vec{x}_{\eta}\right) + C\left(\vec{x}_{\zeta\zeta} + \frac{R}{J^2C}\vec{x}_{\zeta}\right) + 2\left(D\vec{x}_{\xi\eta} + E\vec{x}_{\xi\zeta} + F\vec{x}_{\eta\zeta}\right) = 0$$

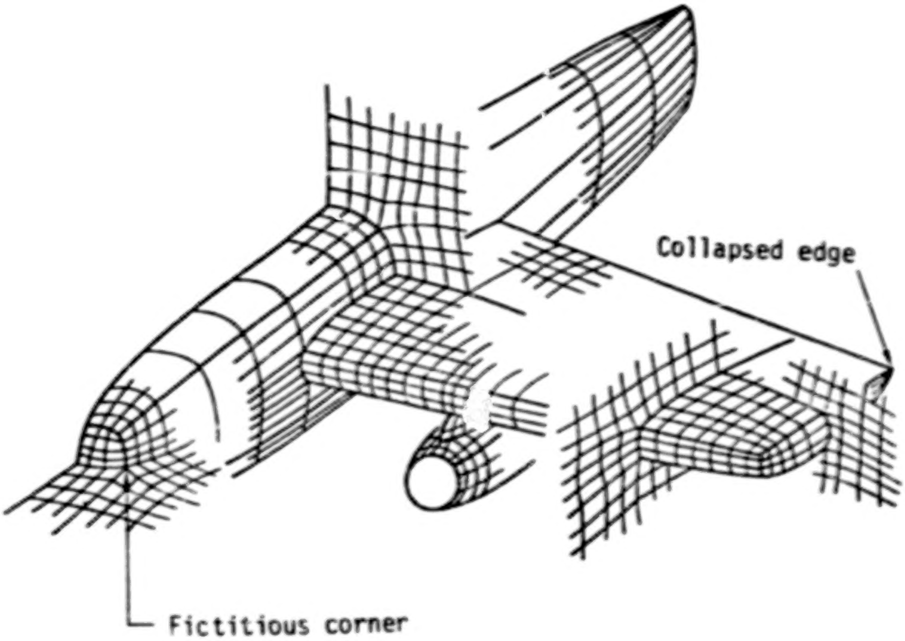
A to F: coupling terms
functions of x, y, and z

P, Q, R: grid control

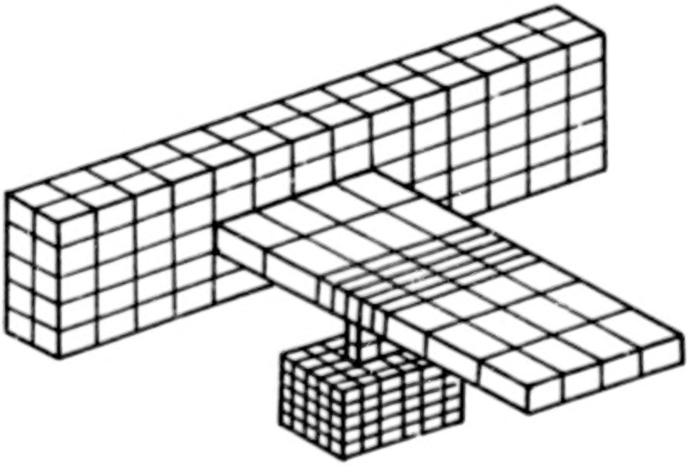
J = Jacobian of the transformation

Figure 1. Block structuring

This is a schematic illustration of a typical block structured grid about a wing/body/nacelle configuration. The multi-block grid obviously provides more desireable grid densities and eliminates the "lost corner." However, it introduces special points termed a "fictitious corner," a "collapsed edge," and a nonanalytic block boundary.



Physical space



Computational space

Figure 2. Comparison of grid structure

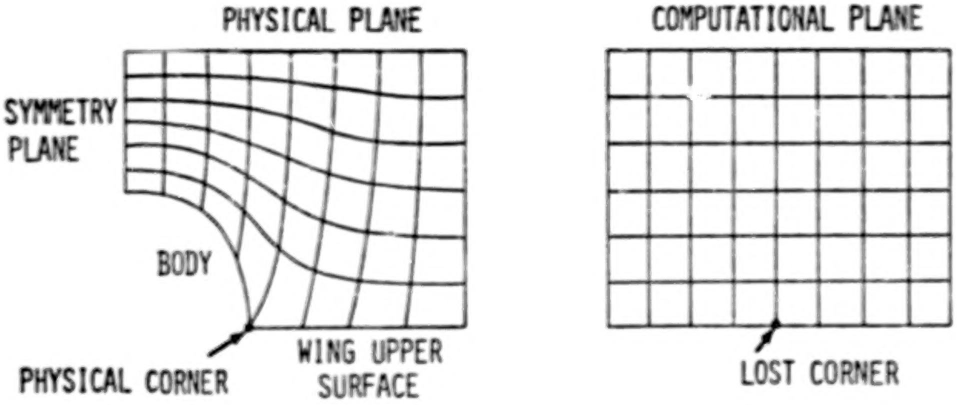
Lost corner - a physical corner transformed into a smooth point in the computational space

Fictitious corner - a smooth point transformed into a corner point in the computational space

Nonanalytic block boundary - grid lines across the block boundary are continuous but not smooth

Collapsed edge (3-D) - grid lines merge together in the physical space

(A) SINGLE-BLOCK GRID


PHYSICAL PLANE
COMPUTATIONAL PLANE
SYMMETRY PLANE
BODY
PHYSICAL CORNER
WING UPPER SURFACE
LOST CORNER

(B) MULTIPLE-BLOCK GRID

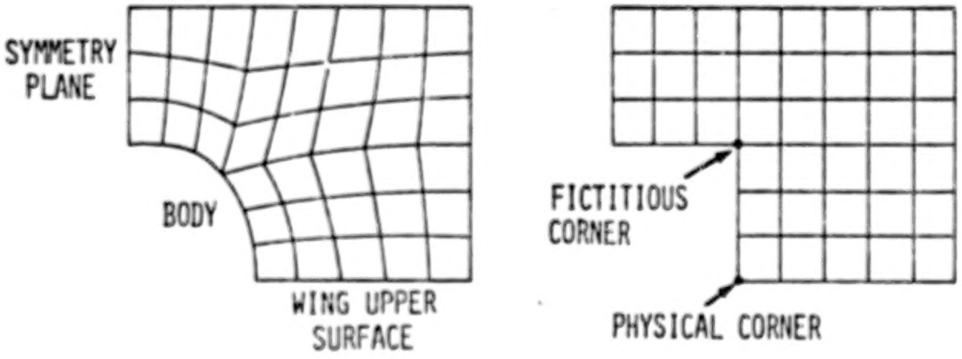

SYMMETRY PLANE
BODY
WING UPPER SURFACE
FICTITIOUS CORNER
PHYSICAL CORNER

Figure 3. Block-structured grid generation process

After defining the overall block structure, a one-dimensional grid generation along the block perimeters produces a perimeter discretization. This provides boundary conditions for a subsequent two-dimensional grid generation producing grids covering the block surfaces. These in turn serve as boundary conditions to produce three-dimensional volume grids filling each block.

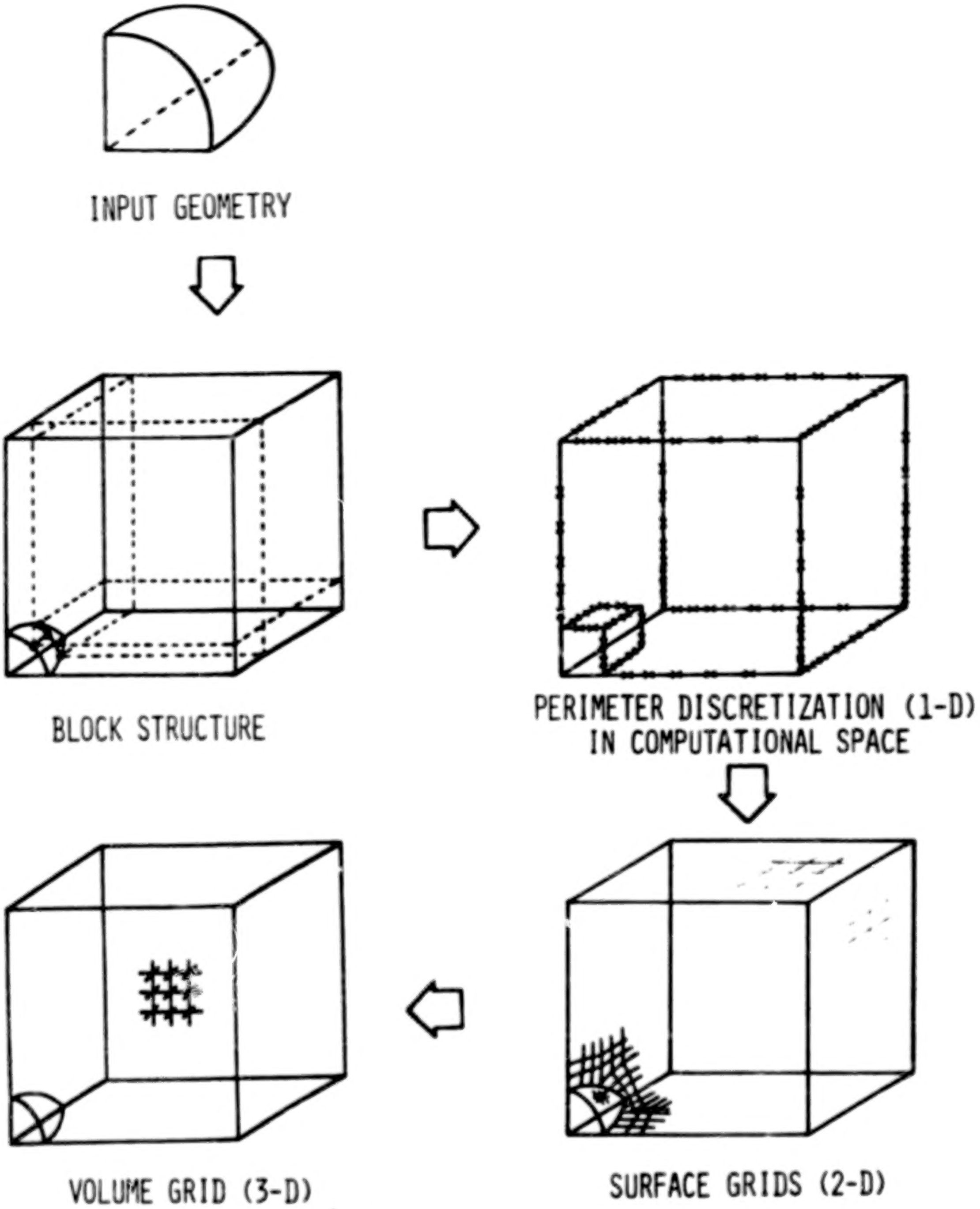
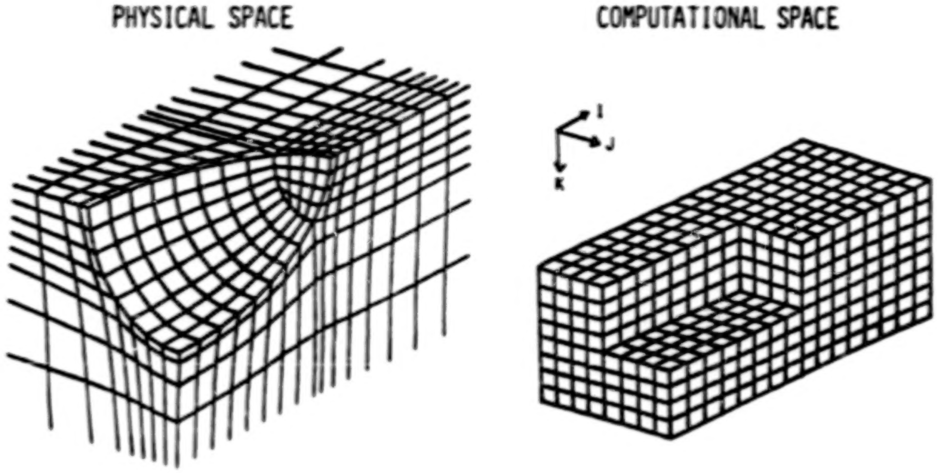

INPUT GEOMETRY
BLOCK STRUCTURE
PERIMETER DISCRETIZATION (1-D)
IN COMPUTATIONAL SPACE
SURFACE GRIDS (2-D)
VOLUME GRID (3-D)

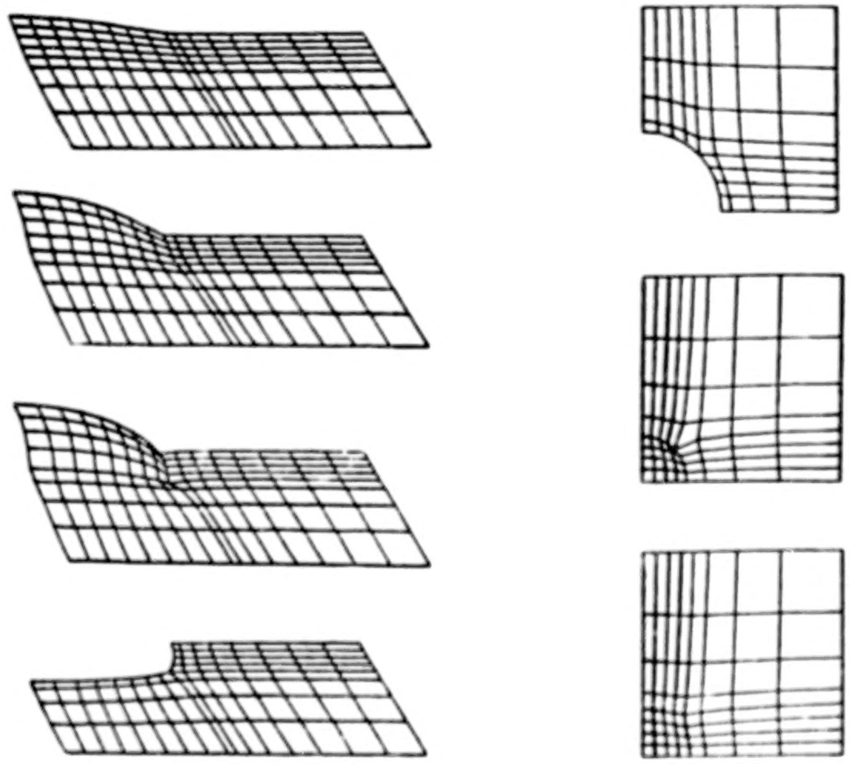
Figure 4. Block-structure grid for an ellipsoid

This example shows the grid around an ellipsoid which has been transformed to a cube in computational space. Fictitious corners can be seen.



(A) ON THE BOUNDARY SURFACE

PHYSICAL SPACE
COMPUTATIONAL SPACE
I
J
K



(B) VOLUME GRIDS: K = CONSTANT

(C) VOLUME GRIDS: I = CONSTANT

Figure 5. Algorithm compatability study

The effect of the grid structure and the special points on the flow solution is explored by solving the potential flow over a cylinder. Cell-oriented flux formulation is used to treat the algorithm issues. Surprisingly, all the grid systems yield good resolution. Accuracy depends on the cell size rather than the grid structure at the special points.

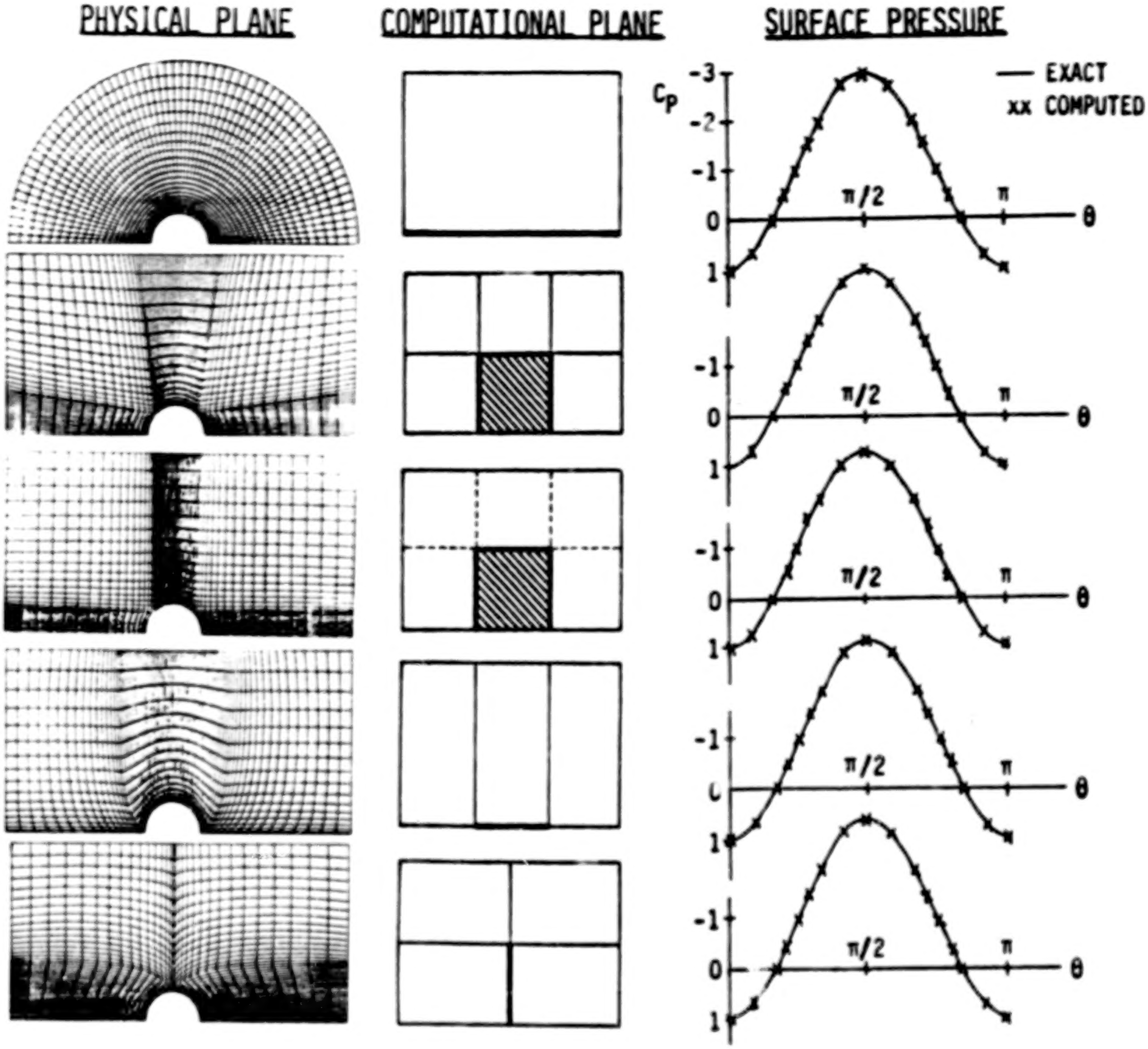
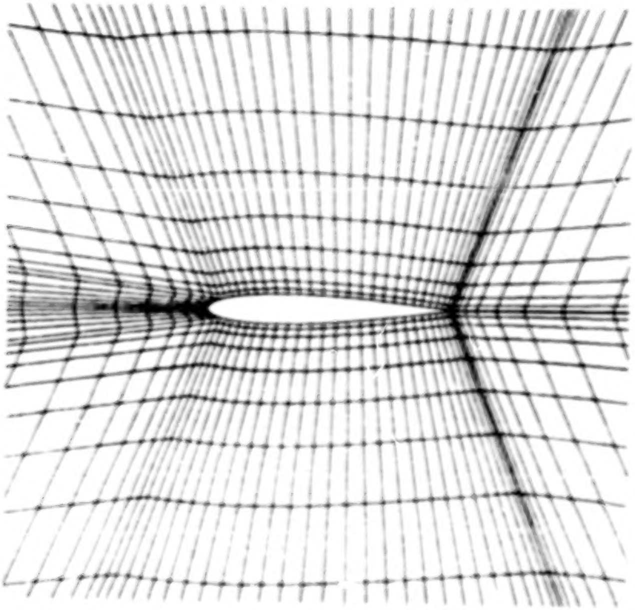


Figure 6. Comparison of grids near an airfoil

The use of multi-block grid is considered for an airfoil. Compared to the ring-type grid, the multi-block grid seems to be overly complex. Its advantage is in its adaptability to more complex geometry.

MULTIPLE-BLOCK GRID



SINGLE-BLOCK GRID

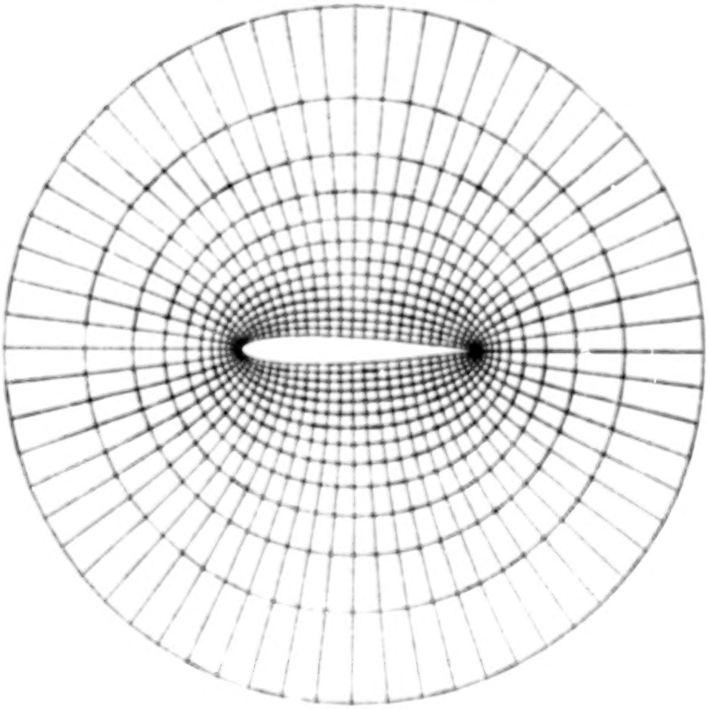


Figure 7. Airfoil study

The ability to produce accurate solutions using the multi-block grid is demonstrated in subsonic and transonic regions. Compared to the results from the ring-type single-block grid, remarkable accuracy was obtained even when the fictitious corner is located in supersonic regions. All the flow and metric quantities are defined at the center of each cell and the artificial density method is adopted for the density retardation in supersonic region.

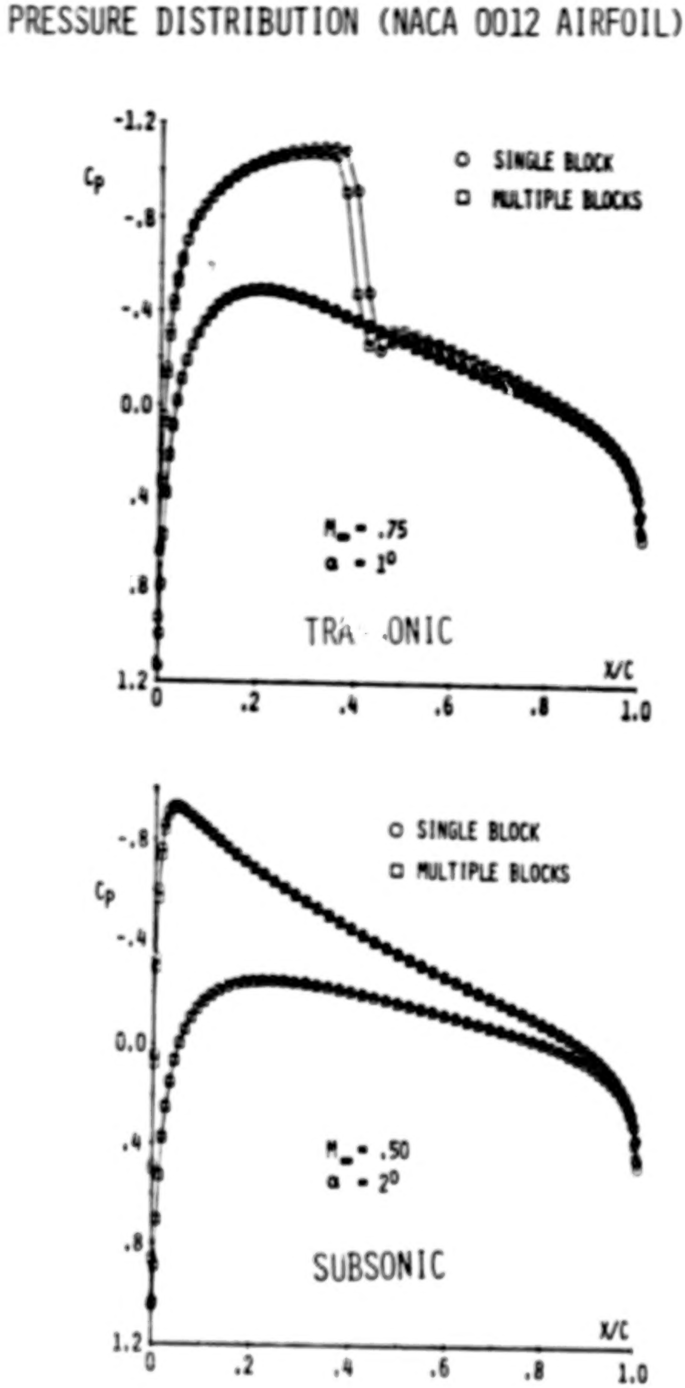


Figure 8. Surface grid for a wing/body
(single-block structure)

The use of the C-type grid provides smooth grid distribution near the wing leading edge. The body surface line on the symmetry plane coincides with a grid line which consists of lost corners. One concern is grid quality at the wing tip.

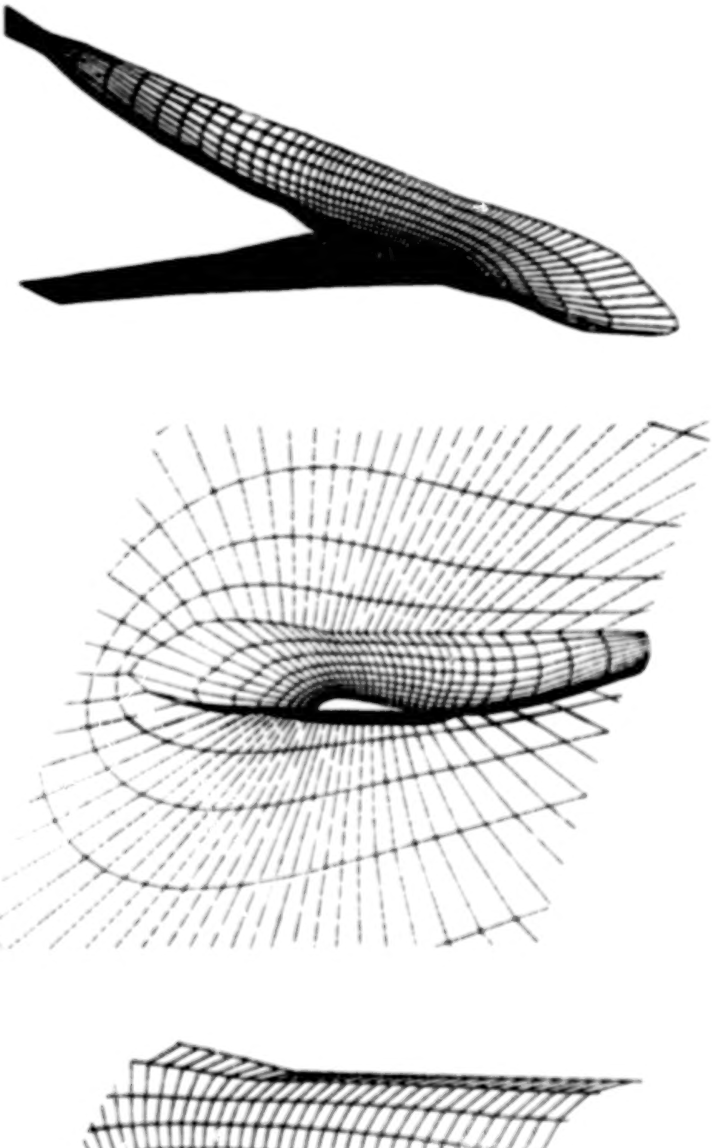


Figure 9. Surface grid for a wing/body
(multi-block structure)

The use of a multi-block grid eliminates the lost corners in the single-block grid of figure 8 and improves the grid quality near the wing tip, while producing the fictitious corners and nonanalytic block boundaries. Its ability to extend to more complex configurations is obvious.

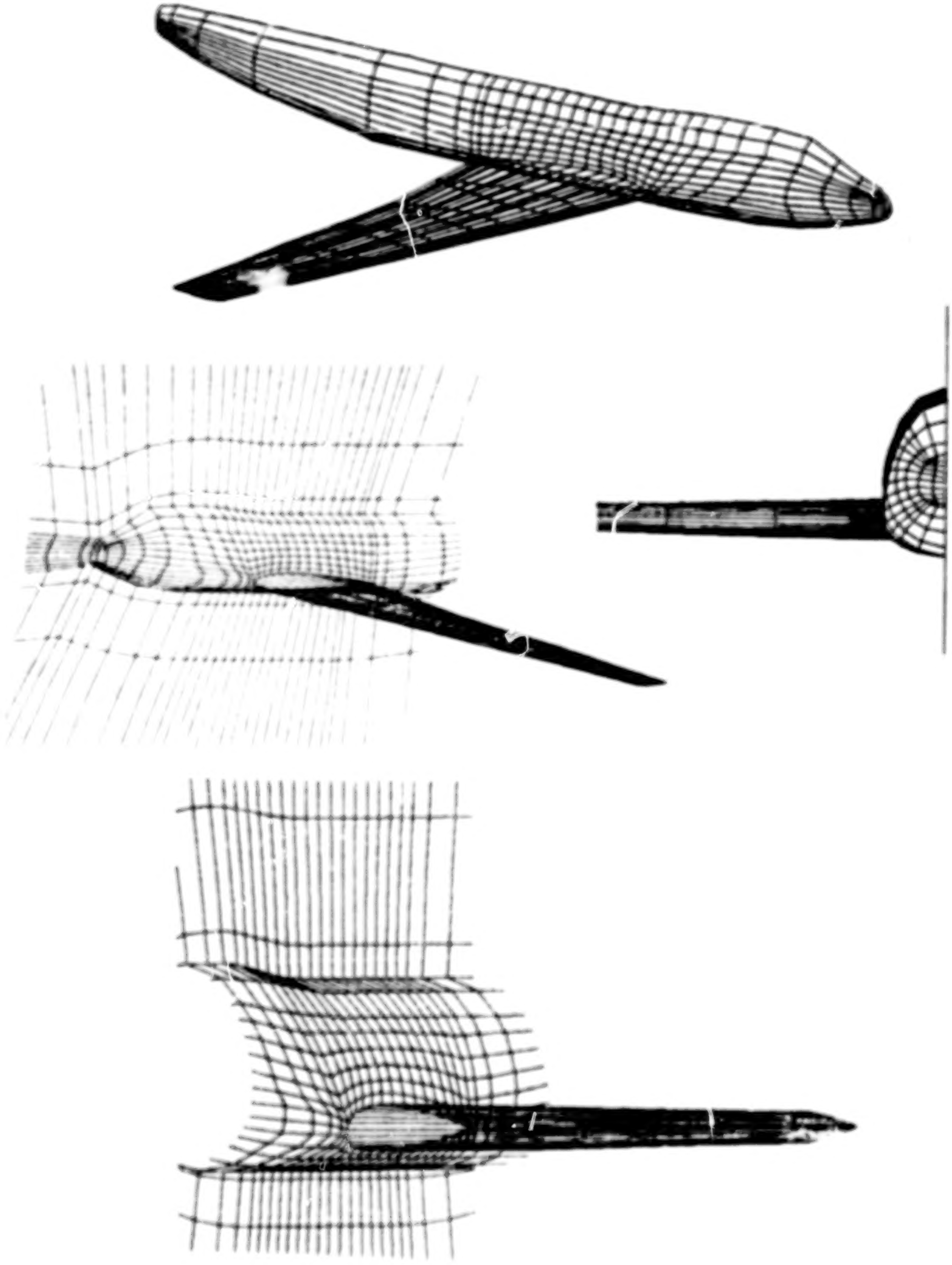
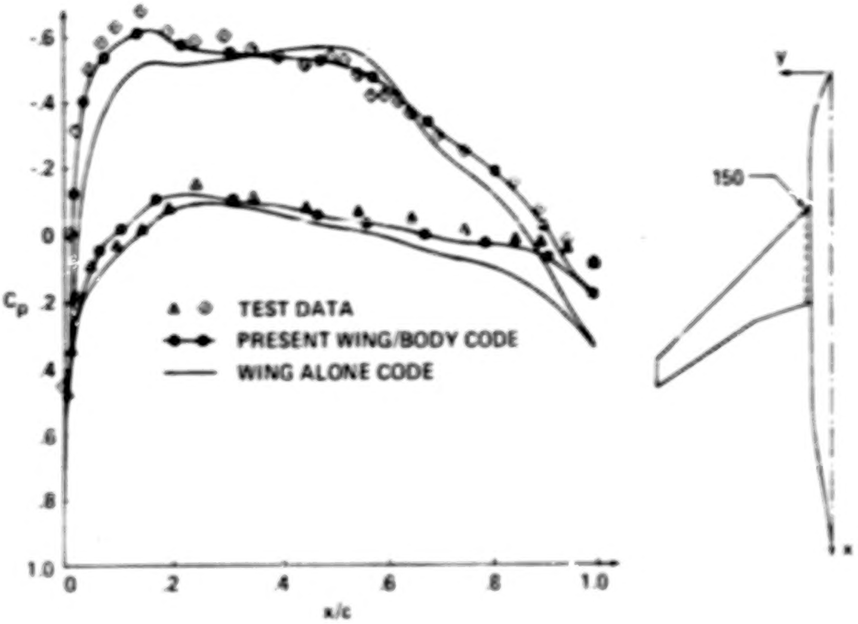


Figure 10. 3-D flow solution

A transonic solution for a wing/body combination is obtained using the single-block grid and compared to the experimental results. The use of body-fitted grid system improves the accuracy near the wing/body junction.

LOW WING CONFIGURATION AT $M_\infty = .84$, $\alpha = 2.8°$



STATION y = 495

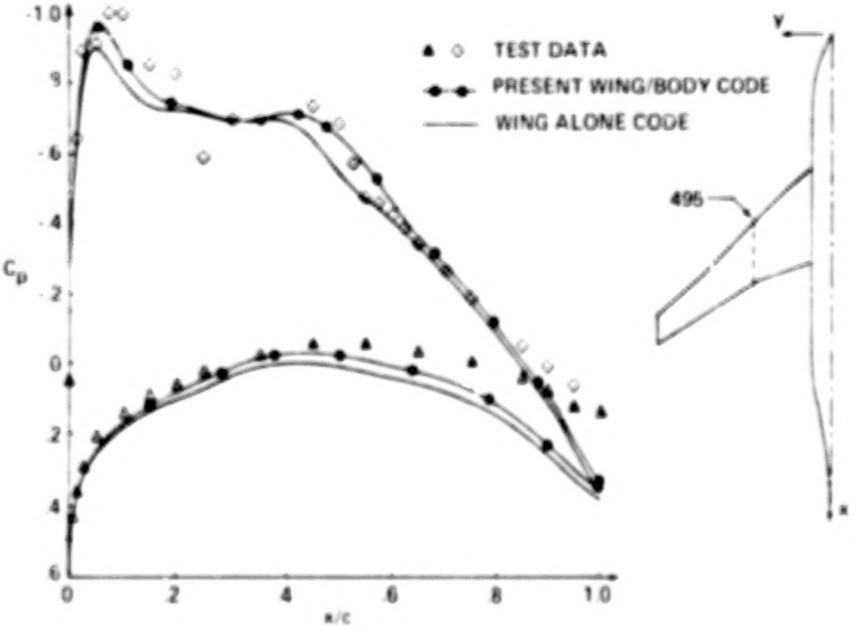
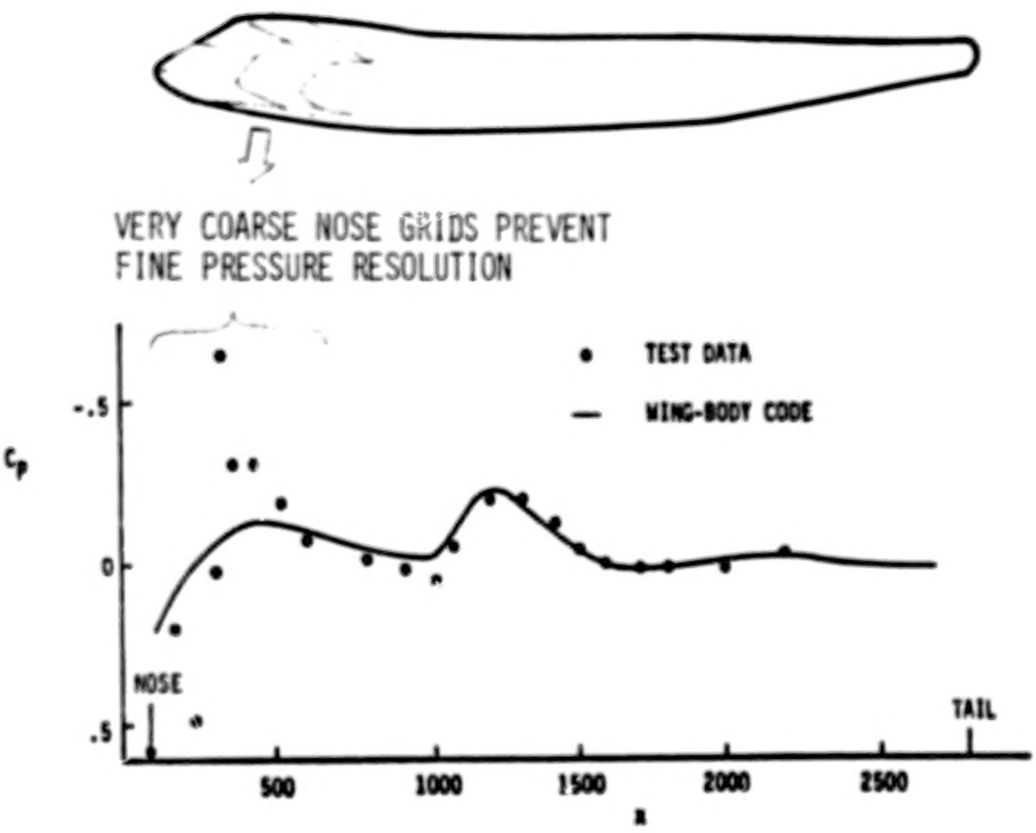


Figure 11. 3-D flow solution

The body-fitted grid system can produce quite accurate pressure distribution even on the body surface. Very coarse nose grid distribution prevents fine pressure resolution in that region.

CROWN LINE PRESSURES FOR 747-200

AT $M_\infty = .84$, $\alpha = 2.8°$



VERY COARSE NOSE GRIDS PREVENT FINE PRESSURE RESOLUTION

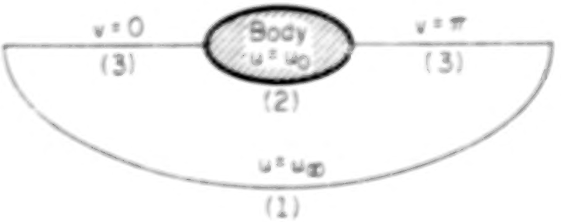
Effect of Grid System on Finite Element Calculation

K. D. Lee and S. M. Yen
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

We have made detailed parametric studies of the effect of grid system on finite element calculation for potential flows. These studies have led to the formulation of a design criteria for optimum mesh system and the development of two methods to generate the optimum mesh system. The guidelines for optimum mesh system are:

1. The mesh structure should be regular.
2. The element should be as regular and equilateral as possible.
3. The distribution of size of element should be consistent with that of flow variables to insure maximum uniformity in error distribution.
4. For non-Dirichlet boundary conditions, smaller boundary elements or higher-order interpolation functions should be used.
5. The mesh should accommodate the boundary geometry as accurately as possible.

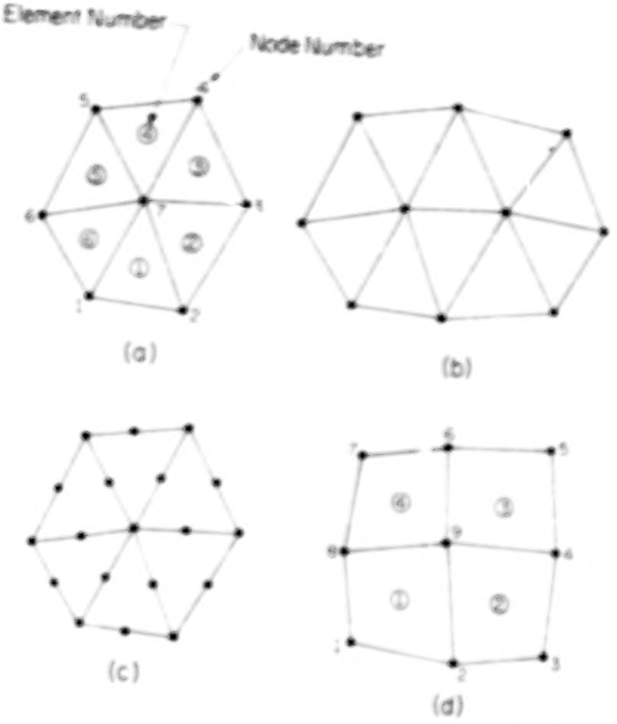
We shall present in this paper the results of our parametric studies.



(u,v): Elliptic-Cylindrical Coordinate System
(Subscript ∞ Denotes Free Stream Condition)

		Problem I	Problem II	Problem III
Type of Boundary Conditions		Dirichlet	Neumann	Mixed
Variable		Stream Function	Velocity Potential	Velocity Potential
Boundary Conditions	at (1)	$\psi = \psi_\infty$	$\phi = \phi_\infty$	$\phi = \phi_\infty$
	at (2)	$\psi = 0$	$\frac{\partial \phi}{\partial u} = 0$	$\frac{\partial \phi}{\partial u} = 0$
	at (3)	$\psi = 0$	$\frac{\partial \phi}{\partial v} = 0$	$\phi + A\left(\frac{\partial \phi}{\partial v}\right) = B$

Fig. 1. We choose three potential flow problems around an elliptic cylinder as the test problems to evaluate and to compare computational errors. In these problems, the computational domain is transformed into a rectangular domain by using the elliptic-cylindrical coordinate system (u,v). This corresponds to an isoparametric element in the physical plane where element boundaries are curved isoparametric lines.



(a)

(b)

(c)

(d)

Fig. 2. Numerical solutions are obtained for the test problems using a sector method. A sector is defined by a combination of elements surrounding a node or nodes. It becomes the finite cut-off zone of influence of the interior node or nodes. The solution procedure is to construct the sector matrix for each sector and to iterate by sweeping all the sectors. This method provides a way to avoid the tedious data handling in constructing the system stiffness matrix and facilitates the treatment of boundary conditions.

Types of Sectors shown are:
(a) Six Triangular Elements, One Interior Node.
(b) Ten Triangular Elements, Two Interior Nodes.
(c) Six Triangular Elements, Seven Interior Nodes.
(d) Four Quadrilateral Elements, One Interior Node.

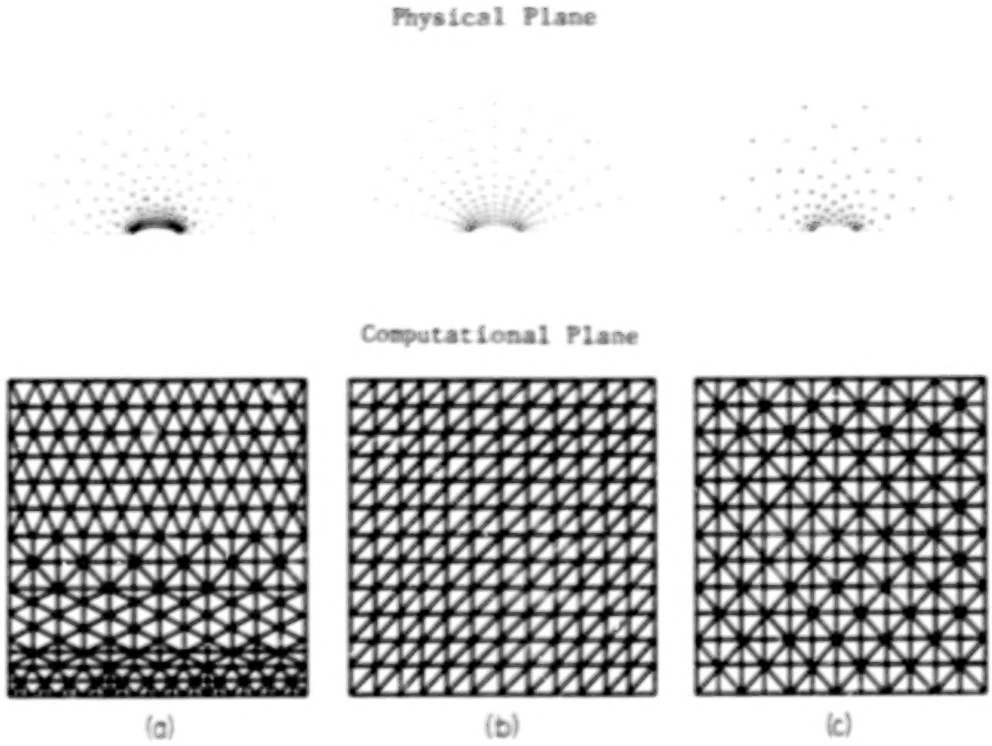


Fig. 3. Three different grid systems for an elliptic boundary. The system (a) has a larger number of nodes and better resolution near the body, however, the structure of the elements is irregular. Table 1 shows the maximum percent error obtained for Problem 1 for the case of Dirichlet boundary conditions. The grid system (a) has much larger error despite the fact that it has more mesh points near the body. This larger error comes from unfair treatment of the influence of neighboring points. The unfair treatment results not only from the irregular shapes of the elements but also from the use of several types of sectors, i.e., sectors consisting of different number of elements. The error increases as more types of sectors are used. The fact that the error in grid system (c) is greater than that in grid system (b) is a further indication of this effect. Only one type of sector, which consists of six elements is used in grid system (b), while two types of sectors, one with eight elements and the other with four, are used in grid system (c). Note that five different types of sectors are used in grid system (a).

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B10
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS
FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE
OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO
GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

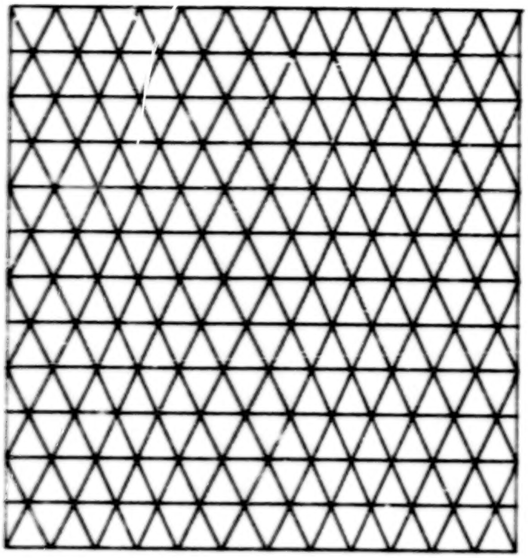
34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT
FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

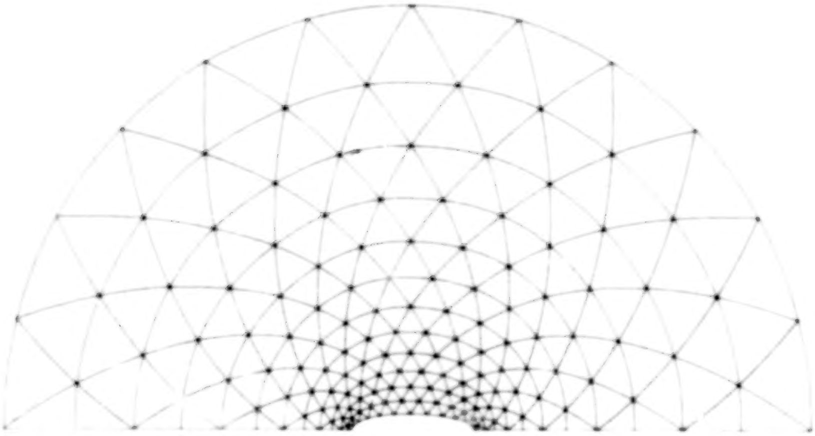
36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL
CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-
SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

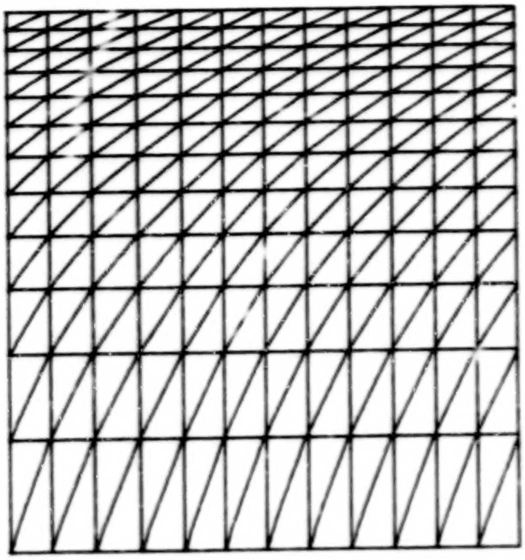


(a) Computational Plane

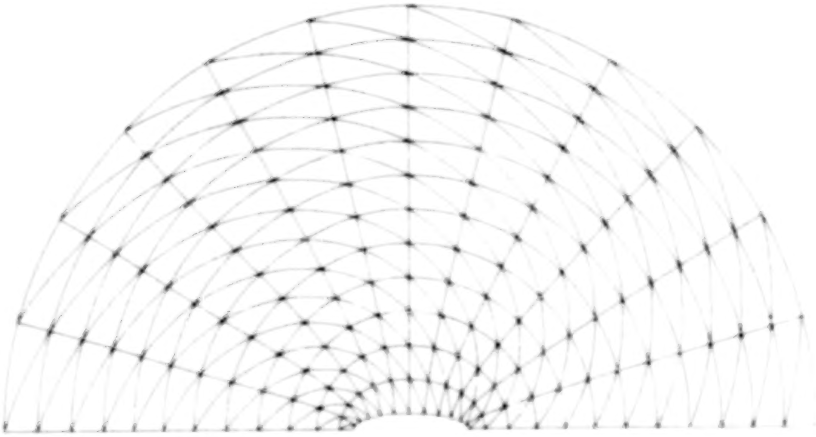


(b) Physical Plane

Fig. 4. The mesh structure of Fig. 3(b) is used to study the effect of element shapes. The shapes considered are equilateral or isosceles triangles, as shown here, in addition to the right-angled triangles, as shown in Fig. 3(b). The maximum errors at both the body surface and the outer boundary are tabulated. The evaluation of the effect of the element shape on the computational errors is based on the comparison of these two errors. For case (1) with right-angled triangles, the error at the body surface is much greater; therefore, the error due to element shape dominates. For case (2) with isoceles triangles, the outer boundary error dominates. For case (3) with equilateral triangles, the two errors are nearly equal. In fact, the error distribution is almost uniform. Such a uniformity in error distribution is important for any flow field computation.

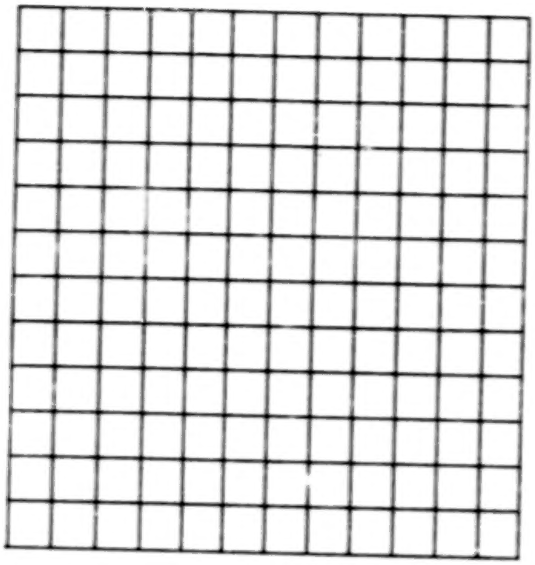


(a) Computational Plane

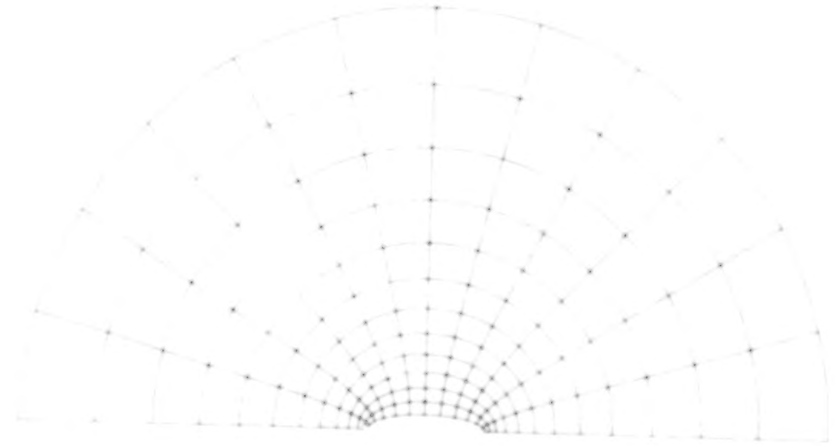


(b) Physical Plane

Fig. 5. The effect of element size distribution was studied by comparing the error for two grid systems shown in Fig. 3(b) and Fig. 5 respectively. These two systems have the same structure; however, the distribution of nodes in the system shown in Fig. 5 is not as uniform. The comparison of errors is given in Table 3. The error for the system of Fig. 5 is greater because the distribution of nodes deviates significantly from the change of field variables.



(a) Computational Plane



(b) Physical Plane

Fig. 6. The error of the system with the triangular element is compared with that with the quadrilateral elements. The interpolation functions in both cases are of second order in the field variables, but differ in their derivatives. The triangular element has a first order accuracy while the quadrilateral element has a second order accuracy. The results are summarized in Table 4. Even though the difference in error in the stream function between the two cases is small, the difference in errors in the velocities is appreciable. In comparing the errors in velocities, it may be more informative to examine the maximum deviations from the exact solutions. This maximum deviation is found to be of $O[10^{-4}]$ per unit free stream velocity for the quadrilateral element and $O[10^{-2}]$ for the triangular element.

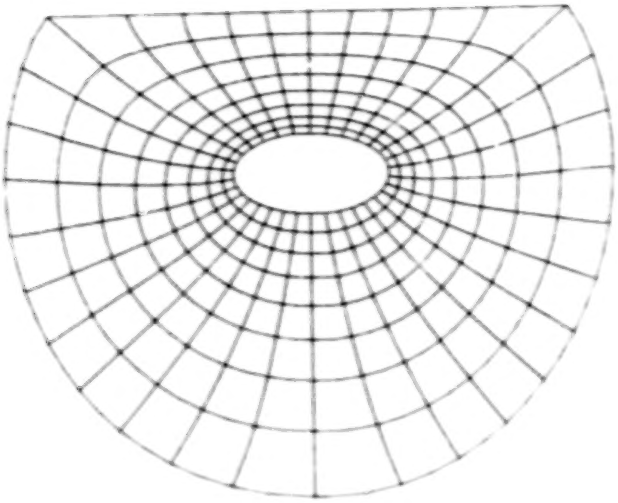


Fig. 7. Optimum mesh system - submerged elliptical cylinder near a free surface.

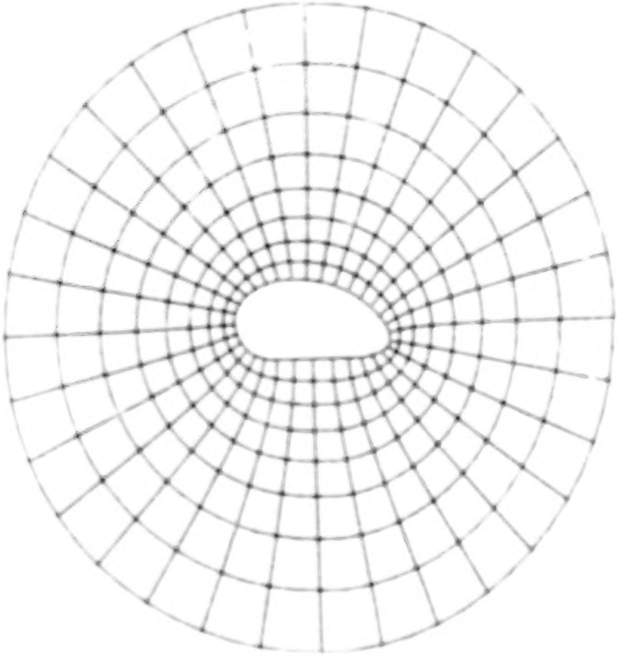


Fig. 8. Optimum mesh system - cylinder of irregular shape.

Two methods of numerical transformation into a set of orthogonal coordinates have been developed to generate an optimum mesh system which meets the guidelines listed above. Figs. 7 and 8 show examples of mesh systems generated.

Table 1. Effect of Mesh Structure

Problem : Problem I, Fig. 1

Element Type : Triangular Element

Mesh System : Fig. 3

Outer Boundary : $u_{out} = u_o + 0.75\pi$

Mesh System	(a)	(b)	(c)
% Error	29.4	0.979	1.64

Table 2. Effect of Element Shapes

Problem : Problem I, Fig. 1

Element Type : Triangular Element

Case	(a)	(b)	(c)
Mesh System	Fig. 3(b)	Fig. 4	Fig. 4
Element Shape	Right-angled Triangles	Isosceles Triangles	Equilateral Triangles
% Error near body	0.979	0.254	0.172
% Error at Outer Boundary	0.363	0.363	0.174
u_{out}	$u_o + 0.75\pi$	$u_o + 0.75\pi$	$u_o + \pi$

Table 3. Effect of Element Size Distribution

Problem : Problem I, Fig. 1
Element Type : Triangular Element
Number of Nodes : 13 x 13
Outer Boundary : $u_{out} = u_o + 0.75\pi$

Mesh System	Fig. 3(b)	Fig. 5
% Error	0.979	6.652

Table 4. Effect of Element Type and Interpolation Functions

Problem : Problem I, Fig. 1
Number of Nodes : 16 x 16
Outer Boundary : $u_{out} = u_o + \pi$

Element Type		Triangular	Quadrilateral
Mesh System		Fig. 3(b)	Fig. 6
% Error	Stream Function	0.856	0.710
	u-Velocity	34.95	1.435
	v-Velocity	33.98	1.096
Maximum Deviation	u-Velocity	0.010	0.0004
	v-Velocity	0.068	0.0004

BLANK PAGE

BLANK PAGE

SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW

C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang
PEDA Corporation
Palo Alto, California 94301

The present paper treats several practical aspects connected with the notion of computation with flow oriented mesh systems. Simple, effective approaches to the ideas discussed are demonstrated in current applications to blown forebody shock layer flow and full bluff body shock layer flow including the massively separated wake region.

The first task in constructing an adaptive mesh is to identify the gross flow structures that are to be captured on the mesh and to work out a grid topology that conforms to them. Among the properties the mesh topology ought to admit are both computational accuracy and algorithmic compatibility. Both these properties are served by grids that feature large connected segments of natural or computational boundaries fitted by mesh surfaces or curves of constant coordinate. But it is neither necessary or always desireable that the entire surface of a particular boundary feature be fitted by a single surface segment of one family of coordinates. For accuracy, convenience, and particularly from the point of view of modern algorithms that embody such features as vector organization, spatial splitting, and implicit solution, it is very desireable that the mesh be composed of identifiable continuous grid lines, not necessarily of homogeneous coordinate type, that run from boundary to boundary.

These notions are illustrated in the application to high Reynolds number full bluff body flow in axisymmetry. Here the basic structure of the turbulent flow is well known, Figure 1. The computational mesh that we have adapted to the flow is shown in Figure 2.

We note that in the mesh shown the computational boundaries — axis of symmetry, bowshock, body, and outflow plane - are all fitted by continuous grid lines. The mesh is so constructed as to be flow aligned over the four principal regions — forebody shocklayer, base recirculation, outer inviscid wake, and inner turbulent viscous wake. We note the wrap around mesh provides continuity of the boundary layer and shear layer in the aft expansion zone. The continuity of the mesh coordinate topology is broken in the recompression zone which embeds a saddle surface of the turbulent flow solution at the interface of the recirculant base flow and downstream viscous wake. The singular topology of the mesh in the base recompression zone is illustrated in Figure 3. The viscous wake core box of the mesh, which provides continuity across the viscous-inviscid wake shear layer, can be regarded as a separate sheet of the topology with a cut taken along a line from the singular point down through the recompression zone to the wake axis.

The cut forms part of a set of construction lines embedded in the mesh, Figure 4. It is central to the method described that these lines which largely define the base mesh structure are also representative of the flow structures which the mesh is to fit. Thus in the approach presented here the construction lines serve the role of supplemental imaginary boundaries along which mesh nodes are distributed according to the usual criteria on ordinary boundaries. The resulting bounded domains can then be filled in with computational grid by any of a large variety of means, for example[1,2,3,4].

The particular grid shown in Figure 2 is quite adequate in concept, though not optimized in detail, and was simply constructed in a single pass using one dimensional distributions along straight coordinate lines between boundary points. Where non-uniform distributions have been required they have been conveniently accomplished using a universal stretching function due to Vinokur[5]. In the program, for the stretching function as we have adapted and use it, the total interval along the coordinate line and the (approximate) first mesh spacings from either end of the interval are specified. The function then returns the distribution between boundary points. As convenient, the stretchings are done variously in X, Y, or S (arc length). The actual X and Y coordinates of mesh points are then found by the functional relationships of points on the given coordinate curve, which of course can be piecewise defined. Where fictitious boundary lines are to be embedded in the mesh, actual boundary points are defined on the connecting coordinate lines at half-first-mesh-cell intervals away from the fictitious lines.

A virtue of meshes constructed of distributions along analytically defined coordinate curves, and particularly straight lines, is that differential displacements of boundary points are readily functionally transformed through kinematic relations into corresponding displacements of the intervening grid points so as to leave invariant the relative distributions of mesh points along the given coordinate curves. For the mesh shown in Figure 2, we presently use this property to analytically deform the outer flow portion of the mesh in relative conformity with the moving, fitted bowshock.

In a similar manner it is intended in future work to differentially adapt the interior base mesh to the changing flow solution by moving the underlying construction lines. A central requirement to do this is to define relationships tying the construction lines to the base flow solution. In this regard it is intended that the X coordinate of the mesh singularity correspond to the axial location of maximum wake pressure. Presumably, the Y coordinate of the singularity which lies on the construction line through the viscous-inviscid wake shear layer ought to be determined from a fit to the axial velocity gradient.

Along the same lines, however, we have developed an adaptive mesh for the blown forebody shock layer which is intended to represent flow over an ablating body. Here we wish to distribute points in predetermined ways in the blown layer, the shear layer interface, and in the outer flow region. In this case a construction line demarking the interface between the blown and outer flow regions can readily and unambiguously be fitted to the zero of the stream function based on mass flux and this is what we have done.

We note in connection with the blown shock layer that the associated flow has regions of steep gradient in density, velocity, mass flux, and temperature and that these properties by no means vary together. We take it that an accurate calculation ought to resolve all these features. Thus we think for this application a mesh distribution approach based on the integral of gradient of a single flow property such as Dwyer[6] has demonstrated is not evidently optimum. A similar distribution based on weighted gradients is certainly feasible but this would appear to be more tedious to implement than a compromise ad hoc distribution tied to key features of the flow structure as we have done. In the paper we shall present curves showing the variation of relevant flow properties across a blown shock layer and show how the simple ad hoc distribution approach we use results in satisfactory resolution of all properties.

References

1. Smith, R.E. and Weigle, B.L., "Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Simulation," AIAA-80-0192, AIAA 18th Aerospace Sciences Meeting, Pasadena, CA, Jan. 14-16, 1980.

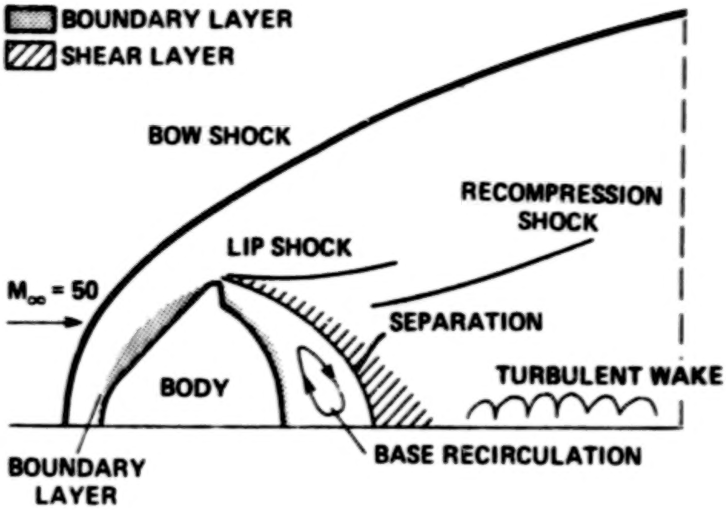
2. Eiseman, Peter R., "Coordinate Generation with Precise Controls," Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford University and NASA-Ames Research Center, June 23-27, 1980.

3. Thompson, J.F., Thames, F.C., and Mastin, C.W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 15, 299, 1974.

4. Middlecoff, J.F. and Thomas, P.D., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA-79-1462, AIAA Computational Fluid Dynamics Conference, Williamsburg, VA, July 23-25, 1979.

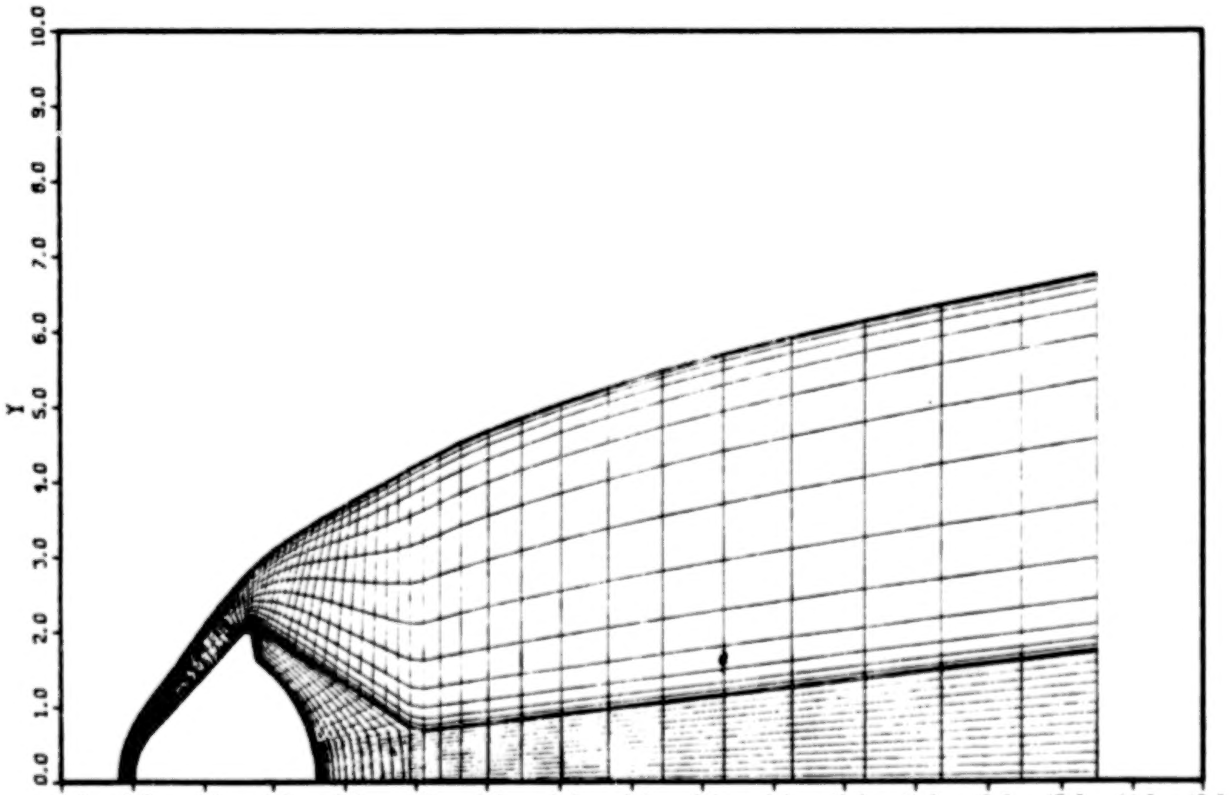
5. Vinokur, Marcel, "On One-Dimensional Stretching Functions for Finite-Difference Calculations," Final Technical Report for Period July 1, 1978 to June 30, 1979, Grant No. NSG 2086, The University of Santa Clara, CA.

6. Dwyer, H.A., Kee, R.J., and Sanders, B.R., "An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," AIAA-79-1464, AIAA Computational Fluid Dynamics Conference, Williamsburg, VA, July 23-25, 1979.



BOUNDARY LAYER
SHEAR LAYER
BOW SHOCK
RECOMPRESSION SHOCK
LIP SHOCK
$M_\infty = 50$
SEPARATION
BODY
TURBULENT WAKE
BASE RECIRCULATION
BOUNDARY LAYER

Figure 1.- Geometry and principal structure of full bluff body flowfield.



Y
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
X
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0

Figure 2.- Full bluff body mesh.

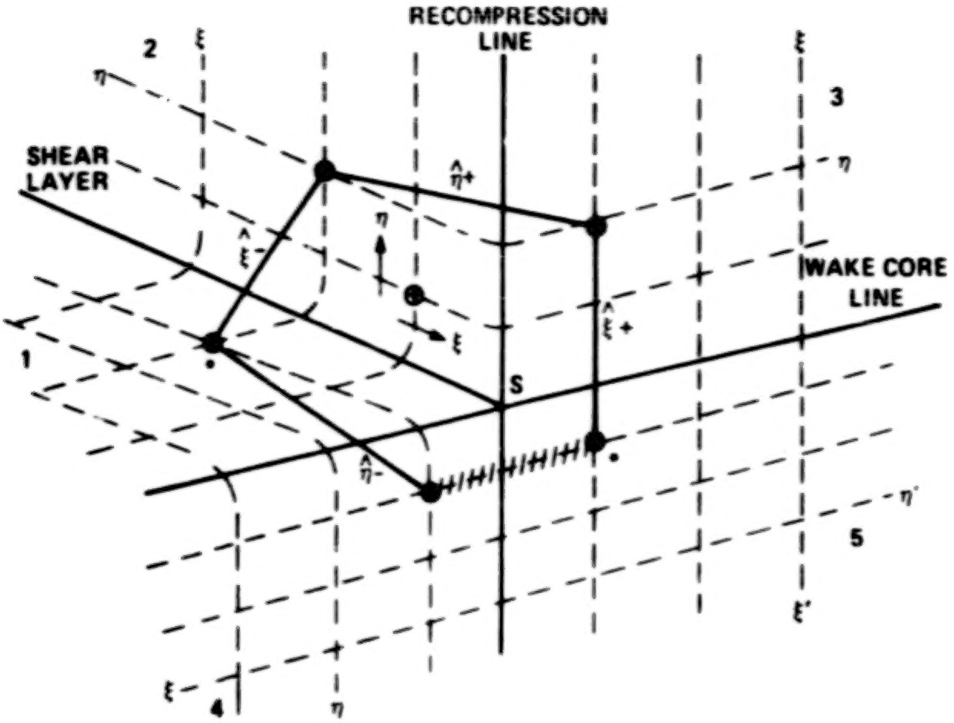

RECOMPRESSION LINE
2
ξ
η
3
SHEAR LAYER
η̂+
ξ̂-
η
ξ̂+
WAKE CORE LINE
ξ
1
S
η̂-
5
η'
ξ'
ξ
4
η

Figure 3.- Base region mesh detail showing topology of the coordinate lines in the vicinity of the singular point s.

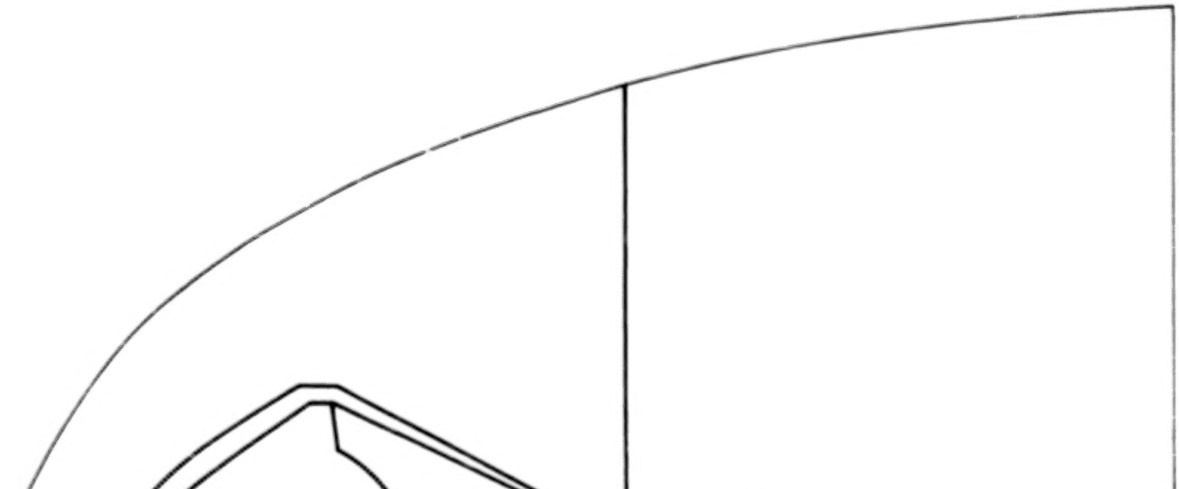

SINGULARITY
VISCOUS WAKE CORE
CUT

Figure 4.- Mesh construction lines.

BLANK PAGE

BLANK PAGE

An Analytical Transformation Technique for Generating Uniformly Spaced Computational Mesh

Youn H. Oh

Hughes Aircraft Company

Abstract

An analytical transformation method which can map arbitrary physical coordinate grid distribution into desired computational coordinate with uniform grid distribution is derived. The transformation function and its higher derivatives are differentiable. Salient features include; 1) precise control of grid sizes, 2) more than one location of clustered grids, 3) exact positioning of particular computational nodes in the physical plane, 4) ensuring several patches of uniformly spaced grids in the physical plane for the higher accuracies (such as at the boundaries), etc., while keeping the variation of grid spacing continuous to avoid numerical instability.

Work was performed while author was employed by Old Dominion University Research Foundation under NASA Research Grant NSG 1087(J. E. Harris, Technical Monitor, High-Speed Aerodynamics Division, NASA LRC)

TRANSFORMATION FUNCTIONS

y = Coordinate in physical plane.

η = Coordinate in computational plane.
Nodes are uniformly spaced.

Parameters Specified	Program	Computed
N = Number of pivot η_{p_i} (i=1, 2, ---N) α_i (i=1, 2, ---N) β_i (i=1, 2, ---N)	⟶ "FIXED" ⟶	$\eta, y, \frac{dy}{d\eta}, \frac{d^2y}{d\eta^2}$

$$f(\eta) = \frac{\partial y}{\partial \eta} = \beta_o + \frac{1}{2}\sum_{i=1}^{N}\left[\beta_i \operatorname{erfc}\left\{\frac{\gamma}{\alpha_i}\left(\eta-\eta_{p_i}\right)\right\} - \left\{\operatorname{sign}\left(\alpha_i\right) + 1\right\}\beta_i\right]$$

$$\begin{aligned} y = \sum_{i=1}^{N} \frac{\beta_i \alpha_i}{2\gamma} \Bigg[& -\frac{\gamma}{\alpha_i}\left(\eta_{min} - \eta_{p_i}\right)\operatorname{erfc}\left\{\frac{\gamma}{\alpha_i}\left(\eta_{min} - \eta_{p_i}\right)\right\} + \frac{1}{\sqrt{\pi}} \\ & \cdot e^{-\left\{\frac{\gamma}{\alpha_i}\left(\eta_{min} - \eta_{p_i}\right)\right\}^2} + \frac{\gamma}{\alpha_i}\left(\eta - \eta_{p_i}\right)\operatorname{erfc}\left\{\frac{\gamma}{\alpha_i}\left(\eta - \eta_{p_i}\right)\right\} \\ & - \frac{1}{\sqrt{\pi}} \; e^{-\left\{\frac{\gamma}{\alpha_i}\left(\eta - \eta_{p_i}\right)\right\}^2}\Bigg] + \left[\beta_o - \frac{1}{2}\sum_{i=1}^{N}\left\{\operatorname{sign}(\alpha_i) + 1\right\}\beta_i\right] \\ & \cdot \left(\eta - \eta_{min}\right) + y_{min} \end{aligned}$$

$$\frac{\partial^2 y}{\partial \eta^2} = -\sum_{i=1}^{N} \frac{\beta_i}{\sqrt{\pi}} \; \frac{\gamma}{\alpha_i} \; e^{-\left\{\frac{\gamma}{\alpha_i}\left(\eta - \eta_{p_i}\right)\right\}^2}$$

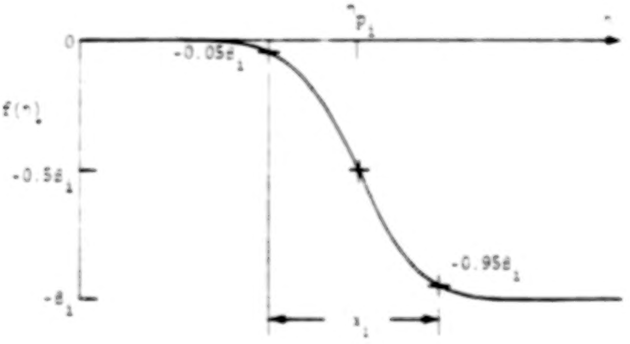
BUILDING BLOCKS OF TRANSFORMATION FUNCTION $f(\eta)$

η_{p_i} (i=1, 2, ---N) = "Pivot points," particular specifiable values of η where $f'(\eta)$ assumes local maxima.

γ = 2.326, a convenient constant for the scaling α_i.

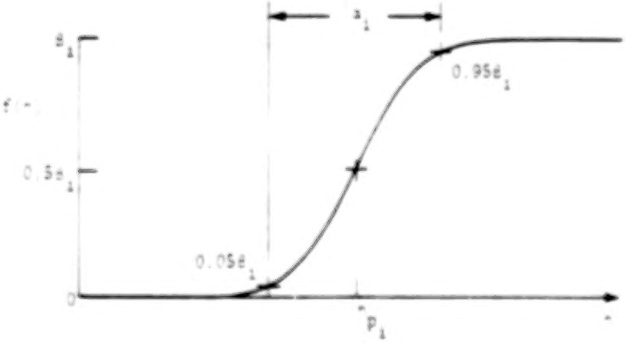
α_i (i=1, 2,---N) = "Width parameter," specifies width in η in which 90 percent of grid size variation takes place around the pivot η_{pi}. $\alpha<0$ specifies increasing grid sizes, and $\alpha>0$ specifies decreasing grid size at pivot η_{pi}.

β_i (i=0, 1,2---N) = step heights for the pivots, which decides the ratio between the sizes of node spacings in y_j on both sides of the pivot.



$$f(\eta) = \frac{\beta_i}{2}\,\mathrm{erfc}\left\{\frac{2.326}{\alpha_i}(\eta - \eta_{p_i})\right\} - \beta_i, \text{ when } \alpha_i > 0$$

(a)



$$f(\eta) = \frac{\beta_i}{2}\,\mathrm{erfc}\left\{\frac{2.326}{\alpha_i}(\eta - \eta_{p_i})\right\}, \text{ when } \alpha_i < 0$$

(b)

THE EFFECTS OF CONTROL PARAMETERS, EXAMPLE 1

Grids are clustered only at η_{min} with a single pivot.

<u>Specified Parameters</u>

Total node points = 51

Number of pivots = 1

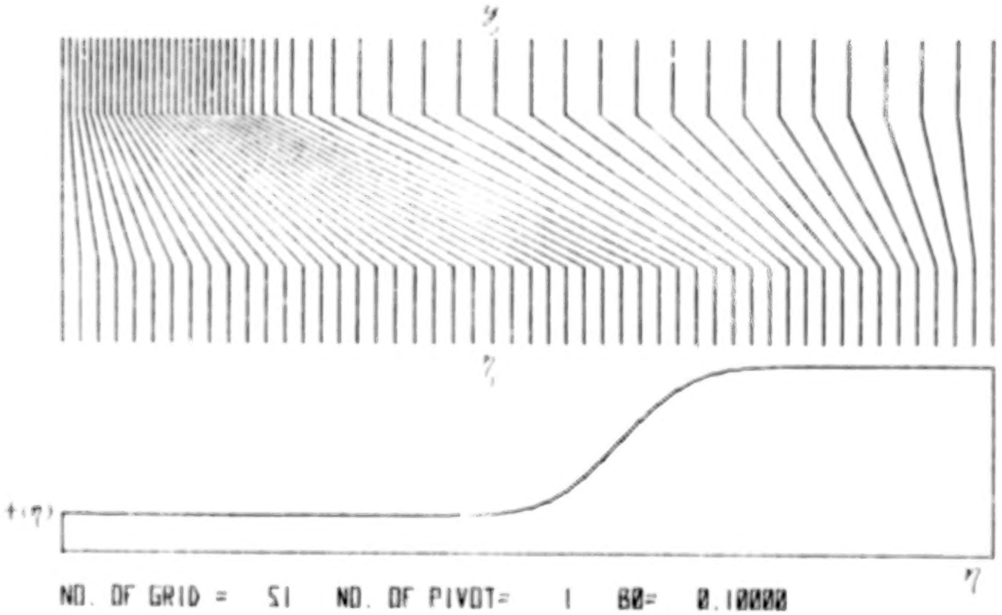
$\eta_{p1} = 0.6$

$\alpha_1 = -0.2$

$\beta_0 = 0.1$

$\beta_1 = 0.4$

Both ends of physical plane nodes have constant spacing.



```
NO. OF GRID =   51   NO. OF PIVOT=    1   B0=   0.10000
S(J)=    0.600
A(J)=   -0.200
B(J)=    0.400
```

EFFECTS OF CONTROL PARAMETERS, EXAMPLE 2

Grids are clustered only at η_{max} with a single pivot.

Specified Parameters

Total node points = 51

Number of pivots = 1

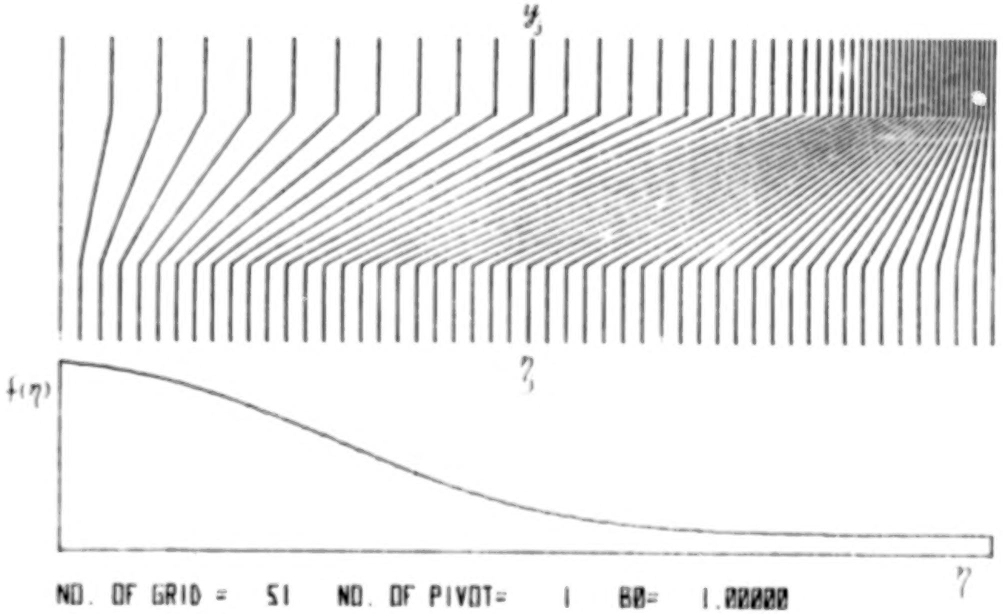
η_{p_1} = 0.3

α_1 = 0.6

β_0 = 1.0

β_1 = 0.9

Grids near y_{max} are uniform but not those near y_{min}.



```
NO. OF GRID =   51    NO. OF PIVOT=    1    B0=   1.00000
S(J)=   0.300
A(J)=   0.600
B(J)=   0.900
```

EFFECTS OF CONTROL PARAMETERS, EXAMPLE 3

Grids are clustered only at an interior point with a pair of pivots.

Specified Parameters

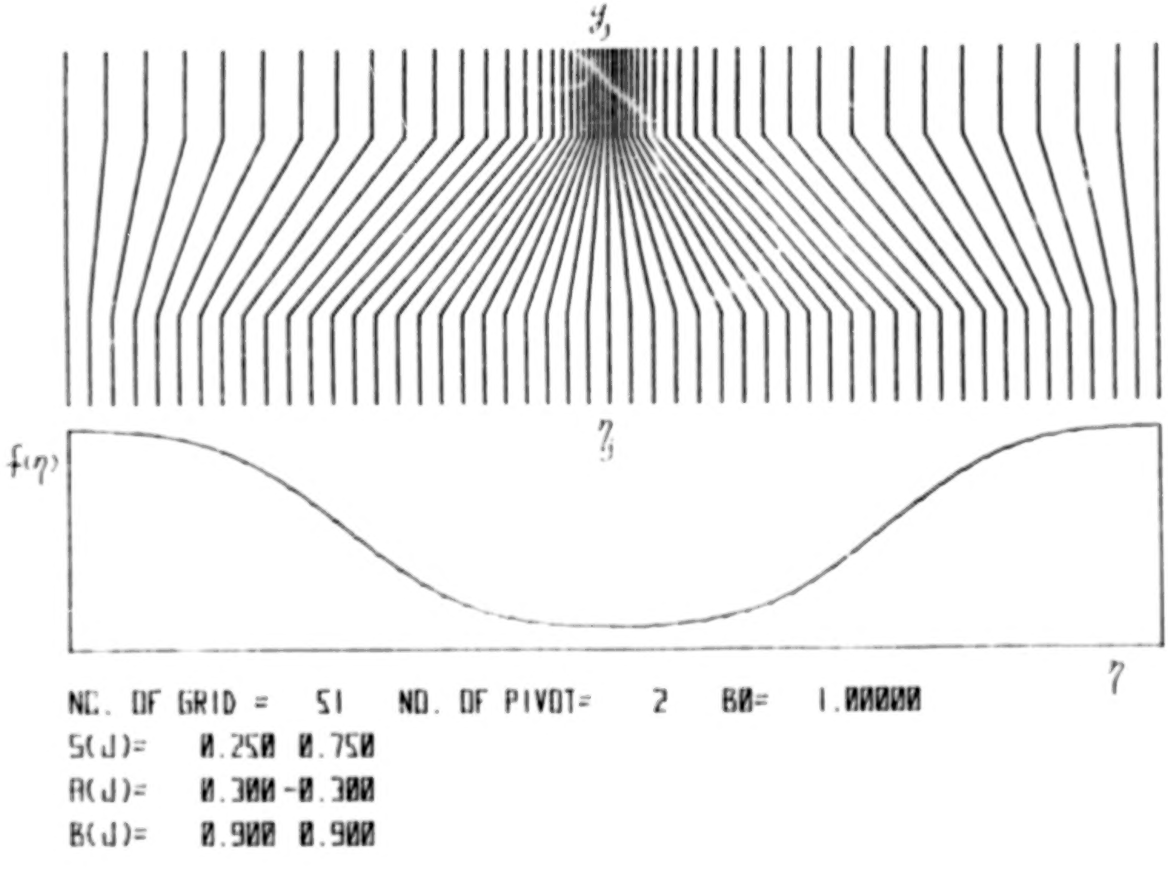
Total node points = 51

Number of pivots = 2

β_0 = 1.0

η_{p_1} = 0.25, α_1 = 0.3, β_1 = 0.9

η_{p_2} = 0.75, α_2 = -0.3, β_2 = 0.9



```
NO. OF GRID =   51    NO. OF PIVOT=    2    B0=   1.00000
S(J)=    0.250  0.750
A(J)=    0.300 -0.300
B(J)=    0.900  0.900
```

EFFECTS OF CONTROL PARAMETERS, EXAMPLE 4

Figure demonstrates smooth variation of grid sizes (large $|\alpha|$ compare to pivot spacing) with multi-pivots.

Specified Parameters

Total node points = 51

Number of pivots = 7

$\beta_0 = 0.05$

$\eta_{p_1} = 0.1, \quad \alpha_1 = -0.14, \quad \beta_1 = 1.$

$\eta_{p_2} = 0.14, \quad \alpha_2 = 0.14, \quad \beta_2 = 1.$

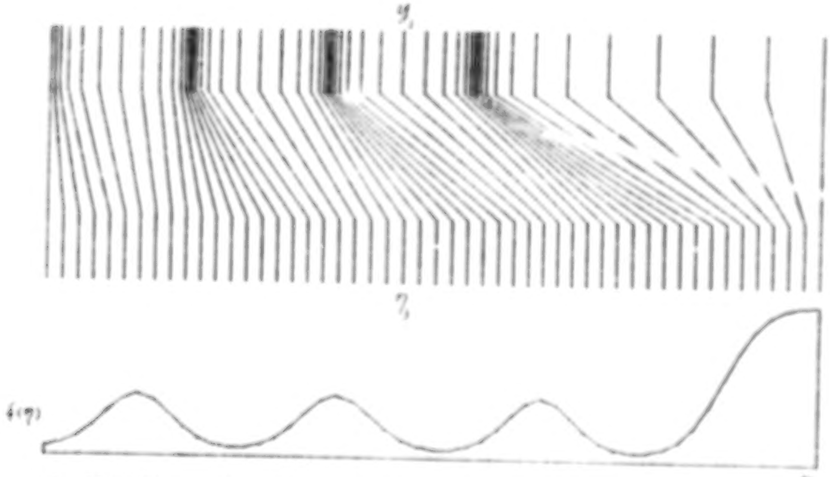
$\eta_{p_3} = 0.36, \quad \alpha_3 = -0.14, \quad \beta_3 = 1.$

$\eta_{p_4} = 0.4, \quad \alpha_4 = 0.14, \quad \beta_4 = 1.$

$\eta_{p_5} = 0.62, \quad \alpha_5 = -0.14, \quad \beta_5 = 1.$

$\eta_{p_6} = 0.66, \quad \alpha_6 = 0.14, \quad \beta_6 = 1.$

$\eta_{p_7} = 0.88, \quad \alpha_7 = -0.14, \quad \beta_7 = 1.$



```
NO. OF GRID =   51    NO. OF PIVOT=    7    B0=   0.05000
S(J)=    0.100 0.140 0.360 0.400 0.620 0.660 0.880
A(J)=   -0.140 0.140-0.140 0.140-0.140 0.140-0.140
B(J)=    1.000 1.000 1.000 1.000 1.000 1.000 1.000
```

EFFECTS OF CONTROL PARAMETERS, EXAMPLE 5

Figure demonstrates discontinuous variation of grid sizes (small $|\alpha|$ compare to pivot spacing) with multi-pivots.

Specified Parameters

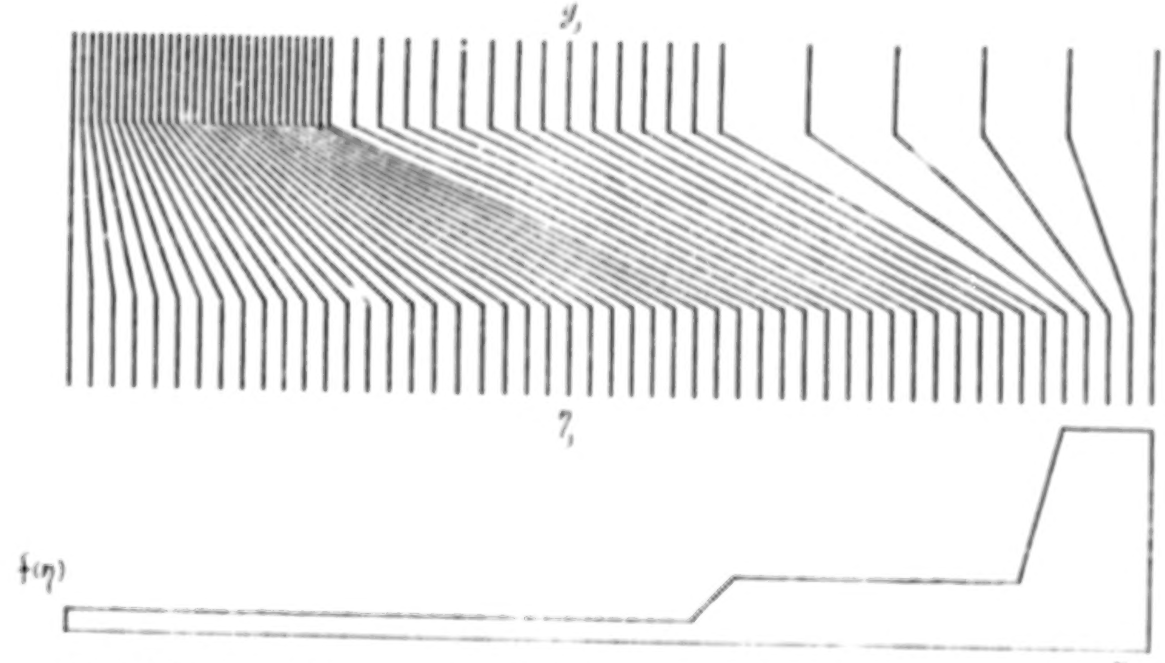
Total node points = 51

Number of pivots = 2

$\beta_0 = 0.1$

$\eta_{p_1} = 0.6$, $\alpha_1 = -0.001$, $\beta_1 = 0.2$

$\eta_{p_2} = 0.9$, $\alpha_2 = -0.001$, $\beta_2 = 0.7$



```
NO. OF GRID =   51   NO. OF PIVOT=    2   B0=   0.10000
S(J)=    0.600 0.900
A(J)=   -0.001-0.001
B(J)=    0.200 0.700
```

CONSTITUTIVE RELATIONS FOR THE DETERMINATION OF β'S

η_{c_k} (k=0, 1, 2, --- K) = "Cluster Points," specifiable values of η which must coincide with particular predetermined values of y_{ck} (k=0, 1, 2, --K) of corresponding k's. The values η_{ck} (hence y_{c_k} also) may or may not necessarily be node points.

ρ_k (k=0, 1, 2, --- K) = Ratio between the immediately neighboring Δy minima and maxima at the cluster point k.

2K simultaneous linear algebraic equations for 2K unknowns, β_0*, β_2, β_3 --- β_{2K} are solved by the program "FIXBY".

---

* $$\beta_1 = \frac{1 - \rho_0}{\rho_C}\beta_0$$

$$(1 - \rho_k)\left[\left\{1 - \text{sign}(a_1)\frac{1 - \rho_0}{\rho_0}\right\}\beta_0 - \sum_{i=2}^{2k-1}\left\{\text{sign}(a_i)\right\}\beta_i\right]$$

$$- \text{sign}(a_{2K})\,\beta_{2k} = 0, \quad (k = 1,2,3,\ldots\ldots K)$$

$$\left\{(\eta_{c_k} - \eta_{min}) + \frac{1 - \rho_0}{\rho_0}\frac{1}{2}\left[\frac{a_1}{\nu}\left\{\theta_1(\eta_{c_k}) - \theta_1(\eta_{min})\right\} - \left\{\text{sign}(a_1)1\right.\right.\right.$$

$$\left.\left.\left. + 1\right\}(\eta_{c_k} - \eta_{min})\right]\right\}\beta_0 + \sum_{i=2}^{2k}\frac{1}{2}\left[\frac{a_i}{\nu}\left\{\theta_i(\eta_{c_k}) - \theta_i(\eta_{min})\right\}\right.$$

$$\left. - \left\{\text{sign}(a_i)1 + 1\right\}(\eta_{c_k} - \eta_{min})\right]\beta_i$$

$$= y_{c_k} - y_{min} \quad (k = 1,2,3,\ldots K)$$

APPLICATION, EXAMPLE 1

The objectives were (1) to transform the physical coordinate y in the range of $0 \le y < 2.5$ to computational coordinate η in the range of $0 \le \eta \le 1.0$;

(2) to obtain grids clustered at $y=0$ and $y=1$ with the same order of minimum grid sizes and the same rates of increase of grid spacing Δy away from $y=0$ and $y=1.0$;

(3) to use a total of 41 nodes where $y=0$, 1.0, 2.5 which are nodes in the physical as well as in the computational plane (i.e., cluster points);

(4) to have grids near $y=2.5$ which are fairly uniform.

Program "FIXBY" is used to compute β_0, β_1, β_2, β_3, with the specified input values;

$\eta_{p_i} = 0.2,\ 0.425,\ 0.825$

$\alpha_i = -0.17,\ 0.17,\ -0.17$ for $i = 1, 2, 3$

$\eta_{c_k} = 0.,\ 0.625$

$y_{c_k} = 0.,\ 1.0$

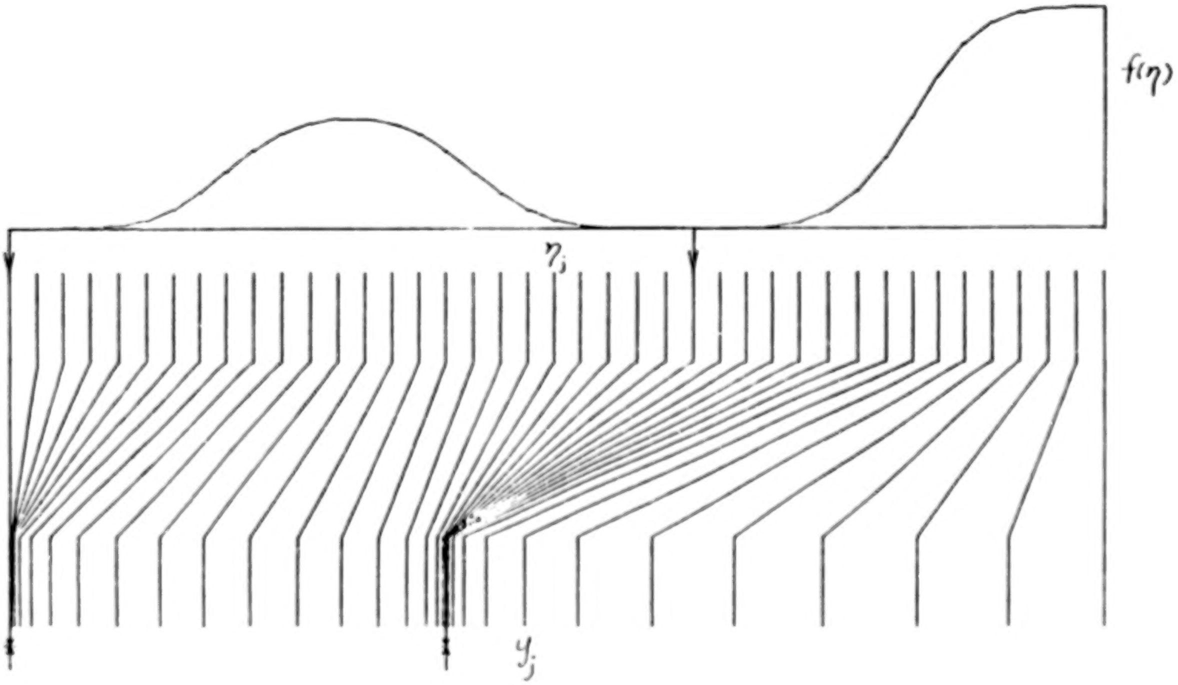
$\rho_k = 0.0025,\ 0.0015$ for $k = 1, 2$

The choice of η_{p_i} decides the way in which the number of nodes are partitioned into different regions and the choice of α_i decides the ratio with which the grid size varies.

Computed β's are

$\beta_i = 0.011071747,\ 4.417627,\ 4.4220557,\ 8.5569778$ $(i=0, 1, 2, 3)$

```
NO. GRID =  41    NO. CLUSTER=  2   NO. PIVOT=  3     B0=   0.011071747
S(J)=       0.200000000   0.425000000   0.825000000
A(J)=      -0.170000000   0.170000000  -0.170000000
B(J)=       4.417627001   4.422055699   8.556977781
```



APPLICATION, EXAMPLE 2

The objectives were to have
(1) a total of 41 nodes;
(2) variable ranges of $0 \le y \le 8.0$ and $0 \le \eta \le 1.0$, respectively;
(3) one interior clustering point at y=0; and
(4) grids near y=0 and y=8.0 which are fairly uniform.

Inputs to program FIXBY were;

$\eta p_i = 0.3, 0.7$

$\alpha_i = 0.17, -0.17 \quad (i=1,2)$

$\eta c_1 = 0.5$

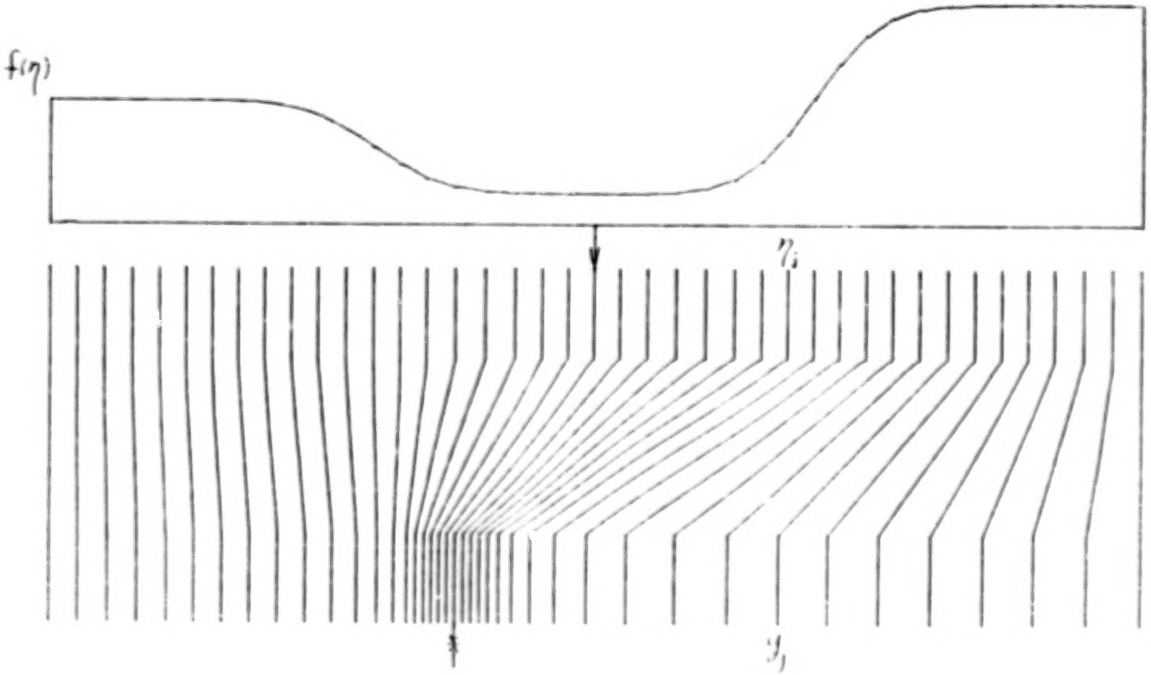
$y c_1 = 3$

$\rho_1 = 0.25$

Computed β's were;

$\beta_i = 8.5714166, 6.4285625, 13.095257 \ (i=0, 1, 2)$

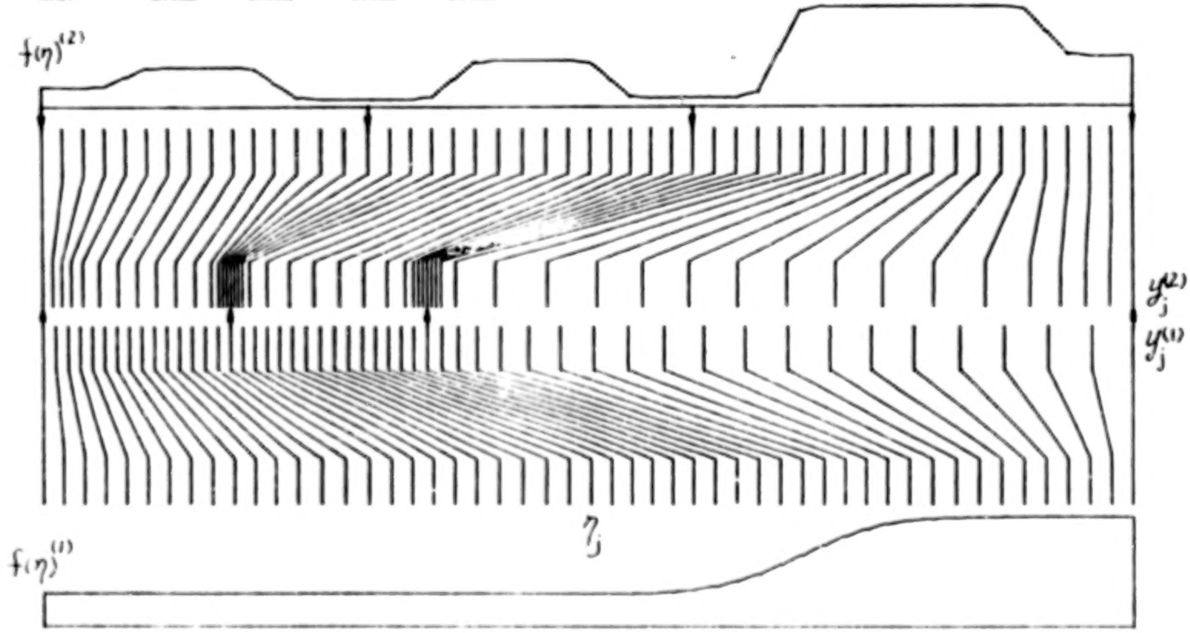
```
NO. GRID =  41    NO. CLUSTER=  1   NO. PIVOT=  2     B0=   8.571416639
S(J)=       0.300000000  -0.700000000
A(J)=       0.170000000  -0.170000000
B(J)=       6.428562480  13.095256998
```



FLOATING GRID CLUSTER REGION

When pivots are sufficiently separated so that there will be a region of "f(η) = constant" between every pivot (small $|\alpha|$ case) is a particular case. A separate program "FLOAT" is programmed for this case. Floating the location of clustering and varying the degree of clustering as well as locating new clustering regions during the computation (of a finite difference solution method) can be conveniently accomplished with this code. Figure shows a transformation of "FLOAT" code with a particular set of input parameters. "FLOAT" redistributed the uniform computational grids η in the physical plane from $y_j^{(1)}$ to $y_j^{(2)}$.

```
NO. OF CLUSTER=     4
A(J)=           1       16       31       51
K(J)=           4        4        4        4
L(J)=           2        2        2        2
H(J)=       0.500    0.200    0.200    0.500
```



```
NO. OF GRID =    51              NO. OF PIVOT=     1              B0=   0.30000
S(J)=         0.700
A(J)=        -0.200
B(J)=         0.700
```

APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION

Samuel Ohring
Computation, Mathematics, and Logistics Department
David W. Taylor Naval Ship Research and Development Center
Bethesda, Maryland

ABSTRACT

The multigrid method (MGM) has been used to numerically solve the pair of nonlinear elliptic equations commonly used to generate two-dimensional boundary-fitted coordinate systems. Two different geometries are considered: one involving a coordinate system fitted about a circle and the other selected for an impinging jet flow problem. MGM uses a nest of grids from finest (upon which the solution is sought) to coarsest and is based on the idea of using relaxation sweeps to smooth the error (equivalent to eliminating high frequency Fourier components of the error). Thus most of the computational work is done on coarser subgrids to eliminate longer wave length components of the error. Two different relaxation schemes are tried: one is successive point overrelaxation and the other is a four-color scheme vectorizeable to take advantage of a parallel processor computer for greater computational speed. Results using MGM are compared with those using SOR (doing successive overrelaxations with the corresponding relaxation scheme on the fine grid only). It is found that MGM becomes significantly more effective than SOR as more accuracy is demanded and as more corrective grids, or more grid points, are used. For the accuracy required here, it is found that MGM is two to three times faster than SOR in computing time. With the four-color relaxation scheme as applied to the impinging jet problem the advantage of MGM over SOR is not as great. Perhaps this is due to the effect of a poor initial guess on MGM for this problem.

The multigrid method (MGM) [1] can numerically solve linear or nonlinear elliptic partial differential equations more rapidly than conventional means of solution such as successive overrelaxation (SOR). MGM can be applied to the numerical solution of partial differential equations not amenable to numerical solution by fast direct matrix solvers such as diagonal decomposition. Thus it was deemed desirable to apply MGM to the numerical solution of the system of nonlinear elliptic equations commonly used to generate boundary-fitted coordinate systems, especially when the number of grid points is large. The standard elliptic equations for a typical mapping, shown schematically in Figure 1, are

$$L_1(x,y) = \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0 \tag{1}$$

$$L_2(x,y) = \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0 \tag{2}$$

where

$$\begin{aligned} \alpha &= x_\eta^2 + y_\eta^2 & \beta &= x_\xi x_\eta + y_\xi y_\eta \\ \gamma &= x_\xi^2 + y_\xi^2 & J &= x_\xi y_\eta - x_\eta y_\xi \end{aligned} \tag{3}$$

and P and Q are functions of ξ and η. Dirichlet conditions are specified on all boundaries of the computational space including the interior slit (which maps to the body in the physical space). Each side of the slit has a set of Dirichlet data with a common value for each of the endpoints of the slit.

The basic idea of MGM is to do most of the computational work on coarser corrective grids containing far fewer points than the finest grid upon which the solution is sought. The grids form a nest, each coarser grid having twice the mesh spacing in each coordinate direction of the previous finer grid. In Figure 2 which represents the Full Approximation Storage scheme of [1]: $u = (x,y)$, $L = \begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix}$ such that Eqs. (1) and (2) become $Lu = F = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$, $1 \leqq k \leqq M$ (k representing the kth grid with M the finest), $\phi = (x,y)$ on the boundaries of the computational space (Dirichlet values so that Λ is an identity operator) and superscripts refer to discretized quantities on the k^{th} grid. (All operations involving ϕ in the flow chart can be ignored, since the Dirichlet conditions are constant on all the grids.) The main idea behind MGM is that relaxation sweeps are a smoothing process which eliminate the highest frequency Fourier components of the error on any grid. First, starting with an initial guess for the solution, several sweeps are carried out on the finest grid to eliminate high frequency components of the error. The smoothed out error is represented by the residual $f^M = L^M U^M - L^M u^M$ and the correction $U^M - u^M$ (where U^M is the exact discrete solution on the finest M^{th} grid). The residual, consisting mainly of longer wave-length Fourier components, is dealt with by solving its coarser-grid approximation

---

[1] A. Brandt, Math. of Comp., Vol. 31, No. 138, April 1977, pp. 333-390.

$$L^{M-1} U^{M-1} - L^{M-1} I_M^{M-1} u^M = I_M^{M-1} f^M \qquad (4)$$

for U^{M-1}, which is represented by $\overline{F}^k$ for $k = M-1$ in the lower right box of Figure 2. The symbol I_k^{k-1} means interpolation of a quantity from the k^{th} grid to the (k-1)st grid. Eq. (4) is solved in the same way as the original equation on the finest grid. If solution of (4) is obtained after several relaxation sweeps, the coarse grid approximation $U^{M-1} - I_M^{M-1} u^M$ to the smoothed out function $U^M - u^M$ is added to u^M. That is $u^M \leftarrow u^M + I_{M-1}^M (u^{M-1} - I_M^{M-1} u^M)$, which is the expression in the lower left box for $k = M$. The new u^M is a better approximation to the solution U^M and is the starting point for more relaxation sweeps for the original set of Eqs. (1) and (2) on the finest grid. If convergence is obtained, the process is complete; if not, the process returns to the coarser grid to sweep the residual equation again. If it doesn't converge after a few sweeps, then the next coarser grid is used to eliminate long wave length errors for the residual equation, etc. Each residual equation has a corresponding residual equation and correction on the next coarser grid. (The residuals were weighted locally as in [1].)

Figure 3 shows computer drawn body-fitted coordinate systems generated to a specific accuracy using MGM and SOR (the two coordinate systems coincide). The relaxation scheme used was successive point overrelaxation. According to the notation used in Figure 1, m and n are 81 and 21, respectively; the slit end points are (ξ_{33},η_{13}) and (ξ_{53},η_{13}), respectively; $(x_\ell, y_b) = (-8.4, -8.0)$

and $(x_r, y_t) = (7.6, 0.0)$; $\Delta x = .2$ and $\Delta y = .4$; $\Delta\xi = \Delta\eta = 1$; and the body is a circle of radius one centered at $(x,y) = (0, -3.2)$. P and Q were set to zero in Eqs. (1) and (2). An experimentally determined, essentially optimum overrelaxation factor of 1.7 was used in the successive point overrelaxation sweeps in both MGM and the SOR method. All coarser corrective grids contain grid points on the slit. The initial guess for $x(\xi,\eta)$, $y(\xi,\eta)$ in the computational space is obtained by extending the Dirichlet data at the outer boundaries throughout the space except at the slit, where the body Dirichlet data are used. The convergence criterion for the solution of Eqs. (1) and (2) was that both L_2-error norms (one for each equation) be less than an input value $||E||_{L_2}$. (This will be called satisfaction of $||E||_{L_2}$.) For Figure 3, $||E||_{L_2} = .001$. To satisfy this criterion, MGM used 32.5 WU and 16.08 CP seconds compared to 66.0 WU and 22.17 CP seconds for SOR. (A work unit (WU) is the equivalent of one SOR sweep on the finest grid, and CP seconds refer to central processor seconds used on the Texas Instruments Advanced Scientific Computer (TI-ASC).) For $||E||_{L_2} = .01$, MGM used 20 WU compared to 29 WU for SOR; CP time was the same for both methods (due mainly to the additional computational work in computing residuals in MGM). The results show that the effectiveness of MGM increases (compared to SOR) as the error norm decreases. This is consistent with the fact that the remaining longer wave length errors are eliminated more slowly using SOR. The parameters $\delta = .3$, $\zeta = .3$ were used to control the flow of MGM. The parameter δ determines the convergence test on each grid and the parameter ζ determines how fast the convergence must be (how fast the high frequency components are eliminated) on each grid. Whenever $\zeta < (||E||^k_{L_2})^{i+1}/(||E||^k_{L_2})^i$ on a k^{th} grid, MGM will then process on the coarser (k-1)st grid with an error norm to be satisfied equal to $\delta(||E||^k_{L_2})^{i+1}$. (Superscripts i,k refer to the i^{th} relaxation sweep and the k^{th} grid, respectively.) These parameters are used as in [1], have a range ($0 < \delta < 1$; $0 < \zeta < 1$), and greatly influence the performance of MGM. The present choice is not necessarily optimum but was the best of a number of choices tried in the unit square.

Figure 4 shows a computer drawn body-fitted coordinate system, similar to Figure 3, generated with MGM and satisfying $||E||_{L_2} = .001$. The grid parameters are (see Figure 1): m and n equal to 129 and 81, respectively;

slit end points of (ξ_{49}, η_{49}) and (ξ_{81}, η_{49}), respectively; (x_ℓ, y_b) = (-7.68, -8.0) and (x_r, y_t) = (7.68, 0.0); $\Delta x = .12$ and $\Delta y = .1$; $\Delta\xi = \Delta\eta = 1$; and the circle of radius one was centered again at (0, -3.2). To satisfy $||E||_{L_2} = .001$ MGM used 21.863 WU and 70.67 CP seconds compared to 102.0 WU and 217.83 CP seconds used by SOR. This represents a significant saving of computer time by MGM. To satisfy $||E||_{L_2} = .01$ MGM used 10.863 WU compared to 17.0 WU used by SOR with CP time essentially the same. These results, along with those for Figure 3, show that MGM is more effective, compared to SOR, when more corrective grids are used and more accuracy is required. Figure 4 has five corrective grids and Figure 3 has three corrective grids (including the finest). The parameters $\delta = .03$ and $\zeta = .2$ controlled MGM for Figure 4. Choosing smaller δ and ζ makes it more likely that all the coarser corrective grids will be used, which is desirable.

Figure 5 shows a computer drawn body-fitted coordinate system generated using MGM and satisfying $||E||_{L_2} = .001$. SOR was also used to generate this grid and is in excellent agreement with MGM. The geometry is motivated by an impinging jet flow problem that is planned to be run on this grid. The flow from the channel interacts with the solid body on the right. The computational space has the same shape as the physical space except that the body is replaced by a slit. Excluding the channel, the grid consists of 137 points in the horizontal direction by 97 points in the vertical direction. The grid for the channel itself consists of 25 horizontal grid points by 33 vertical grid points. The slit (and body) are 49 grid points long. Corner points on the body and channel have been excluded from the grid. Exponential grid spacing was used along various parts of the horizontal and vertical boundaries of the grid. In an attempt to preserve this boundary spacing in the grid interior non-zero P and Q were used. Although grid lines are still bent near the boundaries, they are not bent as much as when P = Q = 0 was tried. To compute this grid (which had 4 corrective grids, including the finest) MGM was "vectorized" on the TI-ASC since it is a parallel processor machine. To accomplish vectorization, which cut computing time by a factor of six, a four-color relaxation scheme was used (i.e., even points of even rows were relaxed simultaneously; odd points of even rows; etc.). With this scheme

MGM used 82.781 WU and 45.57 CP seconds to satisfy $||E||_{L_2} = .001$ when using an overrelaxation factor (RF) of 1.8 on the finest grid and relaxation factors of 1.6, 1.4, and 1.2 for the progressively coarser grids. (Varying RF in this way improved MGM's performance.) SOR (with the four-color scheme) used 170.0 WU and 76.68 CP seconds using a relaxation factor of 1.8, which is about optimum for this SOR. MGM used 60.641 WU and 36.67 CP seconds to satisfy $||E||_{L_2} = .001$ when RF's of 1.6, 1.4, 1.2, 1.0 were used on progressively coarser grids (with 1.6 used for the finest grid). With these RF's MGM used 26.016 WU to satisfy $||E||_{L_2} = .01$ compared to 82.0 WU used by SOR with RF = 1.8. The parameters $\delta = .05$, $\eta = .95$ were used for MGM which was divergent for $\eta < .9$. MGM should perform better with a better initial guess than used here. (The horizontal straight lines in the initial guess were discontinuous at the right-most boundary.)

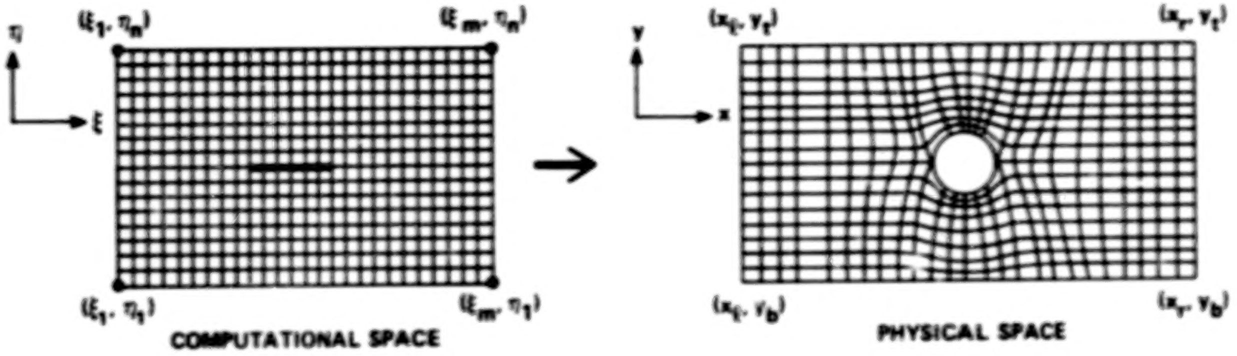

η
ξ
(ξ_1, η_n)
(ξ_m, η_n)
(ξ_1, η_1)
(ξ_m, η_1)
COMPUTATIONAL SPACE
y
x
(x_l, y_t)
(x_r, y_t)
(x_l, y_b)
(x_r, y_b)
PHYSICAL SPACE

Figure 1

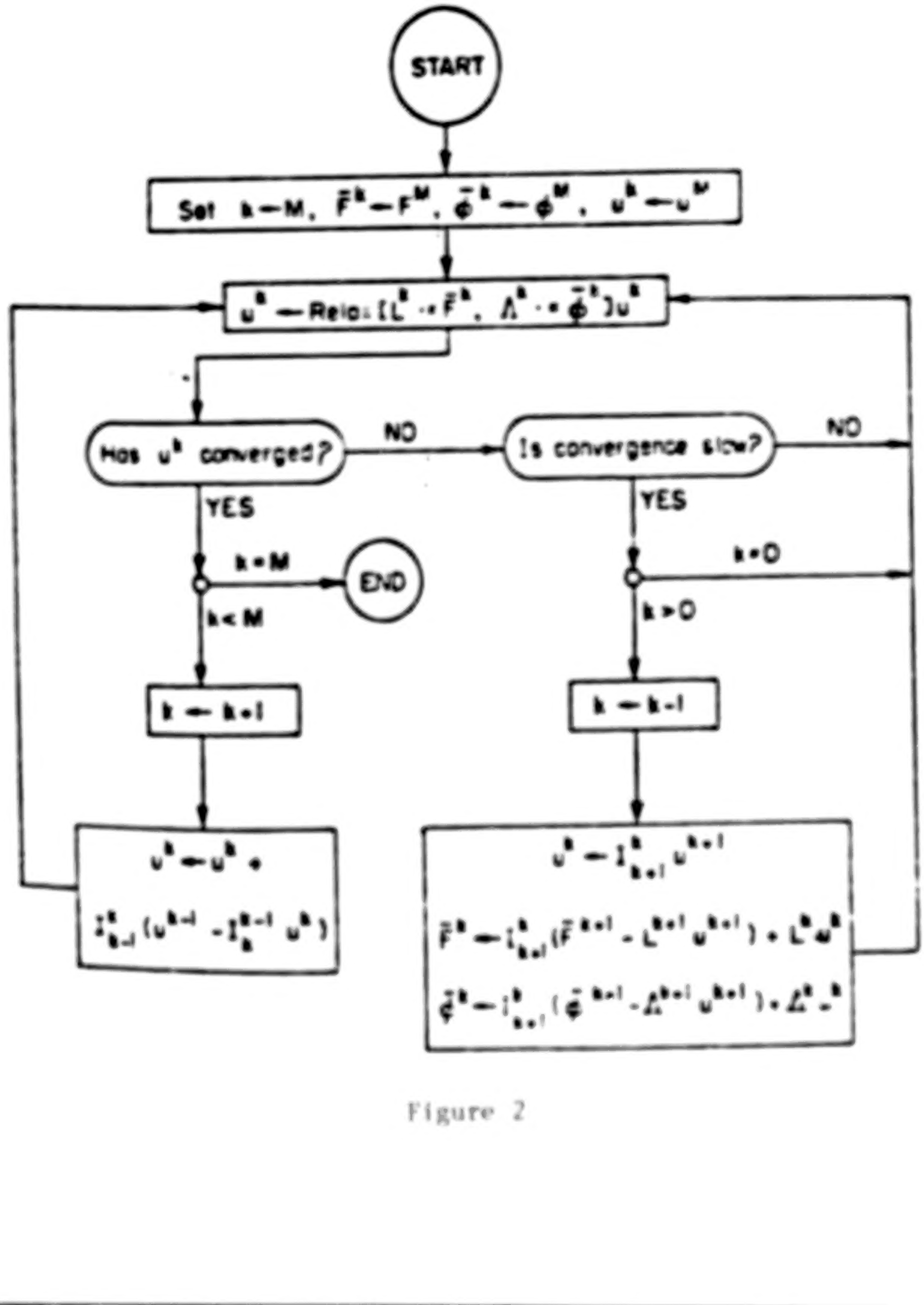

START
Set $k \leftarrow M$, $\bar{F}^k \leftarrow F^M$, $\bar{\phi}^k \leftarrow \phi^M$, $u^k \leftarrow u^M$
$u^k \leftarrow \text{Relax}[L^k = \bar{F}^k, \Lambda^k = \bar{\phi}^k]u^k$
Has u^k converged?
NO
Is convergence slow?
NO
YES
$k = M$
END
$k < M$
YES
$k = 0$
$k > 0$
$k \leftarrow k+1$
$k \leftarrow k-1$
$u^k \leftarrow u^k +$
$I^k_{k-1}(u^{k-1} - I^{k-1}_k u^k)$
$u^k \leftarrow I^k_{k+1} u^{k+1}$
$\bar{F}^k \leftarrow I^k_{k+1}(\bar{F}^{k+1} - L^{k+1}u^{k+1}) + L^k u^k$
$\bar{\phi}^k \leftarrow I^k_{k+1}(\bar{\phi}^{k+1} - \Lambda^{k+1}u^{k+1}) + \Lambda^k u^k$

Figure 2

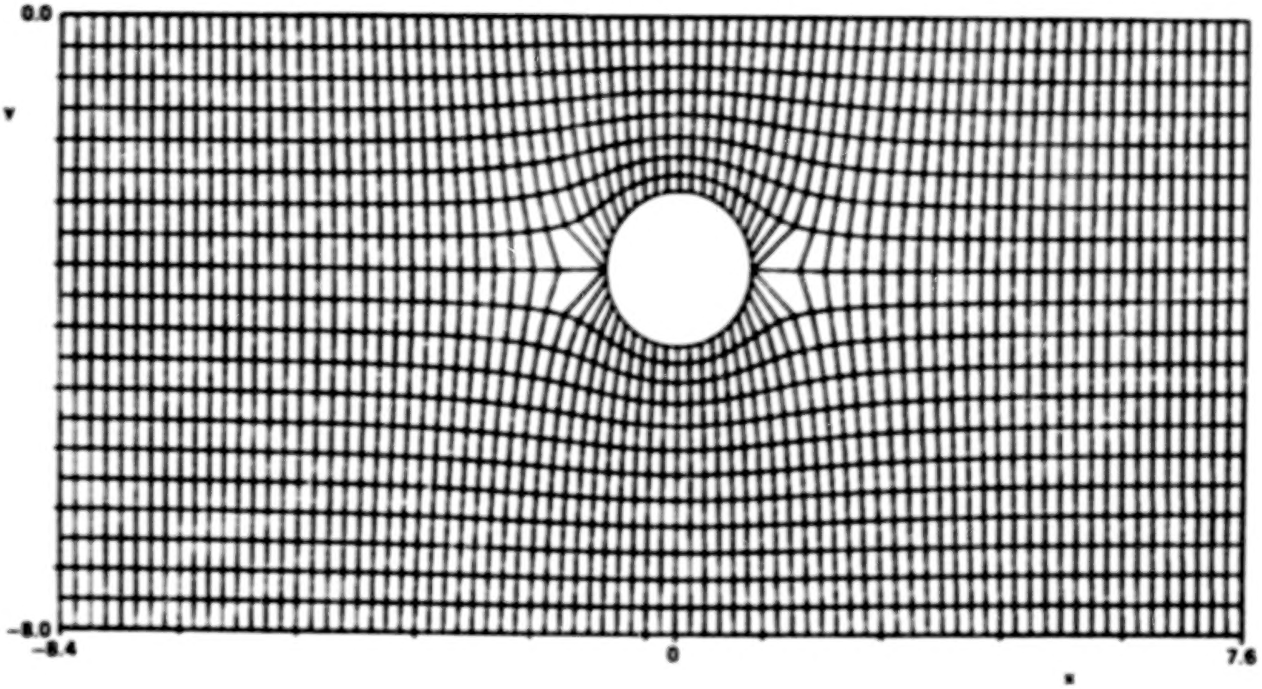

0.0
y
-8.0
-8.4
0
7.6
x

Figure 3

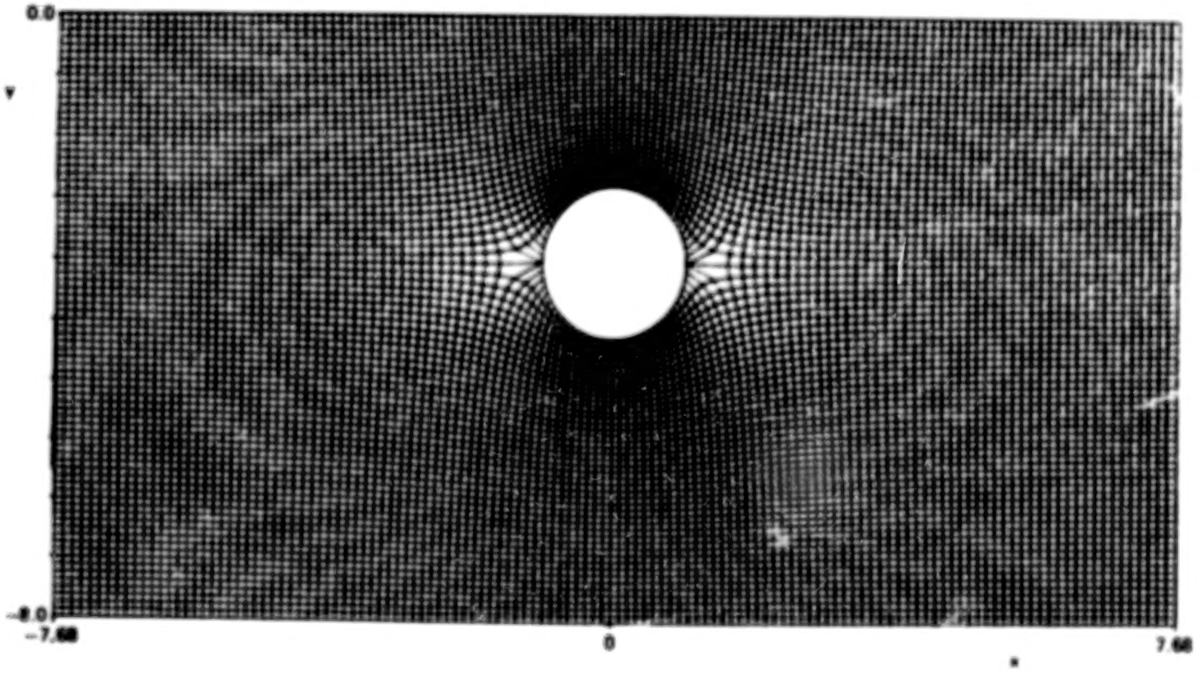

0.0
y
-8.0
-7.68
0
7.68
x

Figure 4

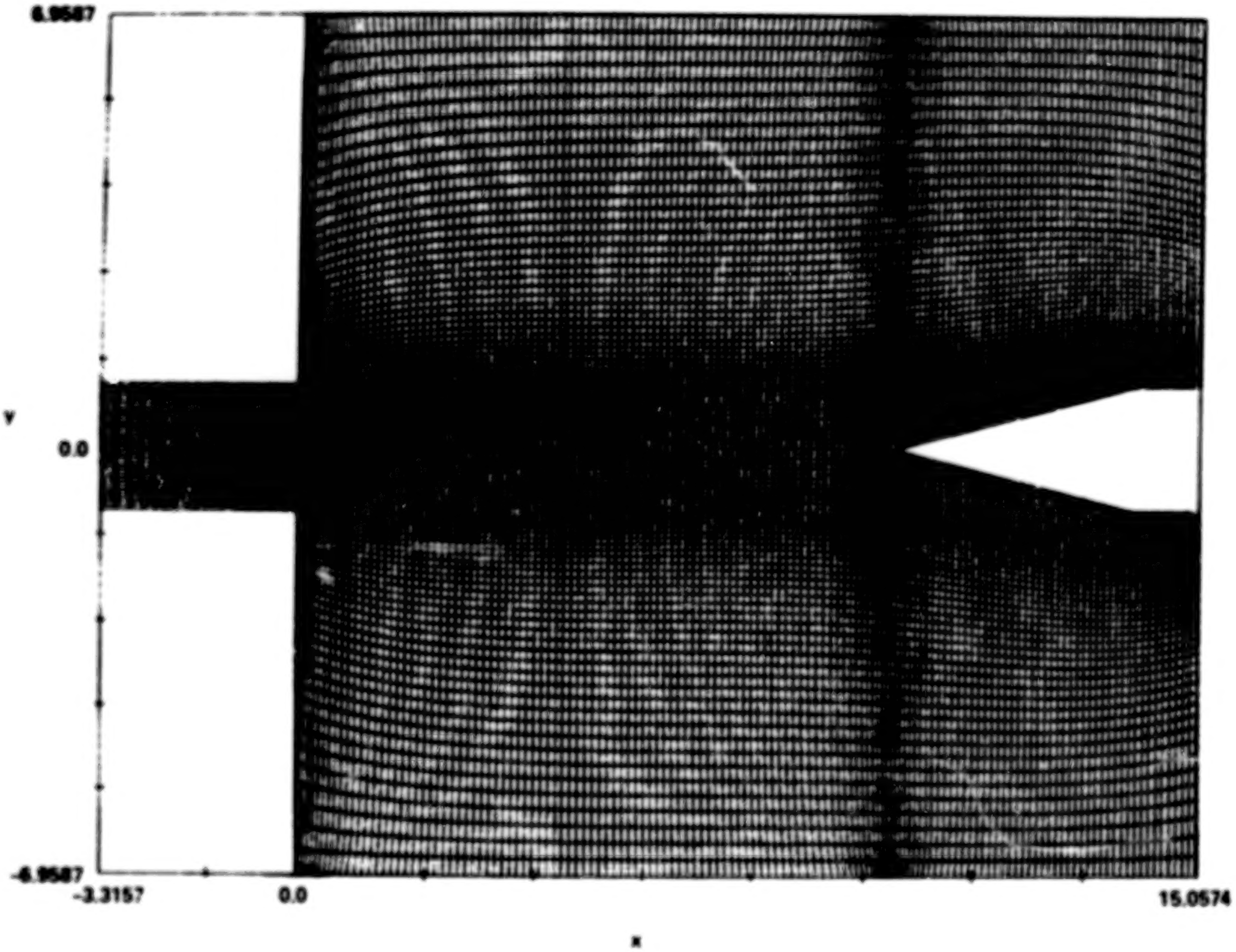

6.9587
y
0.0
-6.9587
-3.3157
0.0
15.0574
x

Figure 5

BLANK PAGE

BLANK PAGE

GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS

Man Mohan Rai* and D. A. Anderson**
Iowa State University
Ames, Iowa

INTRODUCTION

Coordinate system selection is an important consideration in the time asymptotic numerical solution of any fluid flow or heat transfer problem. In solving such transient problems, the physical domain is usually transformed into a rectangular region with boundaries coincident with the physical boundaries. Once this transformation is completed, the transformed equations of motion are integrated until steady state is attained.

Most methods of generating systems of coordinates used in numerical solutions have been developed for elliptic problems. In these methods, the physical domain boundaries are known and the coordinate mesh is determined initially. Generally, the geometry of the mesh is not changed during the computation. Probably the most well known of these methods is the one developed by Thompson et al. (1) in which the transformed coordinates are obtained as a solution of Laplace's equation in physical space. A number of other investigators (2, 3, 4) have developed schemes which can be used to generate appropriate coordinate systems using the same general idea.

---

*Graduate Student

**Professor, Department of Aerospace Engineering

Unfortunately, the solution of a separate elliptic equation is not conveniently included in the solution of a time-dependent set of equations. Hindman et al. (5) solved the two-dimensional time-dependent Euler equations with a truly adaptive grid scheme. The grid motion in time was generated by taking the time derivative of the governing differential equations of the coordinate mapping which was the same as that developed by Thompson. This provided the necessary grid speed equations which were then integrated to obtain the grid motion as a function of time. Hindman's work did not consider techniques which might be used to modify the location of the interior points depending upon the local solution. The interior point motion depended solely upon boundary motion.

A technique for locating mesh points according to local flow information was presented by Dwyer et al. (6). This technique is similar to that used by Olson (7) and involves redistributing the mesh points at the end of any number of integration steps. This method does not permit a simple time integration of a differential equation similar to the equations of gas dynamics for the motion of the mesh points. It is the purpose of this paper to introduce a new technique which provides a simple way of moving the mesh points in physical space and reduces the error in the solution relative to that obtained using a fixed mesh.

Pierson et al. (8) have also worked on the generation of grids which minimize error, but their technique involves the solution of a minimization problem. The extension of such a method to higher dimensions with the accompanying increase in the number of mesh points is not feasible due to the large amounts of computer time necessary to solve minimization problems. The method to be discussed in this paper is very simple in application and takes only a fraction of the time necessary to solve a minimization problem.

THE METHOD

To describe the basic idea employed in this paper, we consider transient problems in one space dimension. Let the physical space coordinates be x and t and let the computational space coordinates be ξ and τ where

$$\tau = t$$

$$\xi = \xi(x,t)$$

We require the calculation of the absolute value of the derivative ($|u_\xi|$) of some representative physical quantity (u) such as velocity, pressure, or temperature and the average value of the same derivative ($|u_\xi|_{av}$) for all mesh points. Given a certain number of grid points, truncation error can be minimized by allocating a number of points to the regions of large gradients and fewer points to the regions of small gradients. For an equispaced grid, a relocation of points in order to minimize error can be carried out. This can be achieved if points at which $|u_\xi|$ is larger than $|u_\xi|_{av}$ attract other points and points at which $|u_\xi|$ is smaller than $|u_\xi|_{av}$ repel other points. In other words, every point induces a velocity at every other point, the magnitude and direction depending upon the local 'excess gradient'. It is logical to assume that the further a point A is from a point B, the smaller the effect of point A on B.

This suggests that a $1/r^n$ law should be used. From the above considerations, it is possible to write

$$(\xi_i)_\tau = K\left[\sum_{j=i+1}^{N} \frac{\left[|u_\xi|_j - |u_\xi|_{av}\right]}{r_{i,j}^n} - \sum_{j=1}^{i-1} \frac{\left[|u_\xi|_j - |u_\xi|_{av}\right]}{r_{i,j}^n}\right]$$

$$i = 2,3\ .\ .\ (N-1) \qquad (1)$$

$$(x_\tau)_i = (\xi_i)_\tau / (\xi_x)_i \qquad (2)$$

where i is the point at which the velocity is being determined, 'N' is the total number of grid points, $r_{i,j}$ is the distance between points i and j in (ξ,τ) space and 'K' and 'n' are constants. The value of K can be determined if the maximum velocity that any point can achieve is specified. Convergence of the grid to a steady-state configuration is obtained by specifying a maximum value for K (K_{max}).

Strong analogies can be found between the present formulation and treating the grid points as point electrical charges whose individual charges are proportional to the local 'excess gradient'. The charges move so as to minimize the quantity

$$E = \sum_{j=1}^{N} \left[|u_\xi|_j - |u_\xi|_{av}\right]^2$$

the minimum value of E being zero.

The collapsing of two computational space points into one physical space point is not possible because of two reasons:

(a) The driving force g,

$$g = |u_\xi| - |u_\xi|_{av} \tag{3}$$

becomes negative when two points get very close and, hence, the points begin to repel each other.

(b) The term ξ_x in Equation (2) gets very large as two points get very close. Hence, for a finite $(\xi_i)_\tau$, $(x_\tau)_i$ tends to zero; i.e., the closer two points get to each other, the more difficult it becomes for them to move toward each other. However, Equation (2) does not prevent extreme stretching of the mesh in physical space, thus giving rise to errors in the calculation of the transformation metrics. The details of preventing extreme stretching for the problems solved in this paper are presented in the section on results.

In the above discussion the driving force g is defined in terms of local and average first derivatives. A better formulation would be one in which g is defined in terms of quantities which are more representative of truncation error. One such quantity is the third derivative of u instead of the first derivative. The appropriate choice depends upon the order of the method being used and the problem itself. The flexibility in choosing the driving force and the quantity to be minimized is a particularly attractive feature of the current scheme.

Two constants K and n appear in Equation (1) and a third one, K_{max} defines the maximum value that K can assume. The constants

K and K_{max} together determine the grid speed. When K is less than K_{max}, the grid speed is determined by K alone and when K is greater than K_{max}, the grid speed is determined only by K_{max}. At present these constants are chosen empirically. In choosing these constants one should bear in mind that very large values of K_{max} result in grid oscillations which in turn result in longer convergence times, and very small values of K_{max} result in low grid speeds and hence, once again longer convergence times are observed. The constant K is calculated by knowing the maximum velocity that any point can achieve in the computational space $\left[(\xi_i)_\tau\right]_{max}$. The rules that govern the choice of $\left[(\xi_i)_\tau\right]_{max}$ are the same as those that govern the choice of K_{max}.

A variation of the constant 'n' between 1 and 8 did not make any difference in the final grid in the one-dimensional case studied and a small difference in the two-dimensional case. The number of iterations for convergence increases slightly when larger values of n are used. However, larger values of n imply a smaller range of influence for any given point. Consider a value of n,

$$n = \frac{2}{\log(2)}$$

When $r = 2$,

$$\frac{1}{r^n} = 10^{-2}$$

This implies that only points adjacent to a given point make a significant contribution to the velocity of that point. Hence, Equation (1) becomes

$$(\xi_i)_\tau \approx K \left| |u_\xi|_{i+1} - |u_\xi|_{i-1} \right| \tag{4}$$

The use of Equation (4) instead of Equation (1) greatly speeds up the grid generation process.

EXTENSION TO MULTIDIMENSIONAL PROBLEMS

The method can be extended to problems in two and three space dimensions without any difficulty. In particular, for a problem in two space dimensions, let the physical coordinates be given by (x,y,t) and the computational coordinates by (ξ,η,τ) where

$$\tau = t$$

$$\xi = \xi(x,y,t)$$

$$\eta = \eta(x,y,t)$$

We now require the calculation $|u_\xi|$ and $|u_\eta|$ for every point and $|u_\xi|_{av}$ for every row of points and $|u_\eta|_{av}$ for every column of points as in Figure 1. The grid speed equations are given by

$$(\xi_{i,j})_\tau = K_1 \sum_{\ell=1}^{M} \left[\sum_{k=i+1}^{N} \frac{\left[|u_\xi|_{k,\ell} - |u_\xi|_{av_\ell} \right]}{r^n} - \sum_{k=1}^{i-1} \frac{\left[|u_\xi|_{k,\ell} - |u_\xi|_{av_\ell} \right]}{r^n} \right]$$

$$(\eta_{i,j})_\tau = K_2 \sum_{k=1}^{N} \left[\sum_{\ell=j+1}^{M} \frac{\left[|u_\eta|_{k,\ell} - |u_\eta|_{av_k} \right]}{r^n} \right. \tag{5}$$

$$\left. - \sum_{\ell=1}^{j-1} \frac{\left[|u_\eta|_{k,\ell} - |u_\eta|_{av_k} \right]}{r^n} \right]$$

$$r = \sqrt{(i-k)^2 + (j-\ell)^2}$$

where K_1, K_2 and n are constants, N the number of points in the ξ direction and M the number of points in the η direction. The values of K_1 and K_2 can be determined by specifying $\left[(\xi_{i,j})_\tau\right]_{max}$ and $\left[(\eta_{i,j})_\tau\right]_{max}$ respectively. Grid convergence can be achieved by specifying $(K_1)_{max}$ and $(K_2)_{max}$ as in the one-dimensional case.

We also have the relationships

$$\begin{aligned} (\xi_{i,j})_\tau &= (\xi_x x_\tau + \xi_y y_\tau)_{i,j} \\ (\eta_{i,j})_\tau &= (\eta_x x_\tau + \eta_y y_\tau)_{i,j} \end{aligned} \tag{6}$$

which yield

$$(x_\tau)_{i,j} = \frac{\left[(\eta_y)_{i,j}(\xi_{i,j})_\tau - (\xi_y)_{i,j}(\eta_{i,j})_\tau \right]}{J}$$

$$(y_\tau)_{i,j} = \frac{\left[(\xi_x)_{i,j}(\eta_{i,j})_\tau - (\eta_x)_{i,j}(\xi_{i,j})_\tau\right]}{J} \tag{7}$$

$$J = \xi_x\eta_y - \eta_x\xi_y$$

From Equation (7) it can be seen that the collapsing of mesh points and the overlapping of grid lines is again prevented as in the one-dimensional case.

Points lying along a constant η line can be made to move tangential to this line by specifying $(\eta_{i,j})_\tau$ to be zero for all these points. A similar procedure can be adopted for constant ξ lines. This facilitates the movement of points along surface boundaries, etc. However, this type of unnatural constraint on the velocity of points leads to a slightly distorted grid as shown in Figure 2. A more natural way of making points move tangential to boundaries is to specify periodic boundaries and use the pseudo points outside the region of interest also to calculate the grid speed. This procedure of calculating the grid speed results in the grid shown in Figure 3. The distortions present in Figure 2 are absent in Figure 3 and the grid is seen to be smooth and uniform. The grids shown in Figures 2 and 3 were generated using a known solution to the two-dimensional transient, linear, viscous Burger's equation.

RESULTS

The first problem solved using the present grid generation technique was the one-dimensional unsteady viscous Burger's equation

$$u_t + uu_x = \mu u_{xx} \tag{8}$$

with the initial condition

$$u(0,x) = \begin{cases} 1 & x = 0 \\ 0 & 0 < x \leq 1 \end{cases} \tag{9}$$

and the boundary conditions

$$\begin{aligned} u(t,0) &= 1 \\ u(t,1) &= 0 \end{aligned} \tag{10}$$

This problem has the steady state solution

$$u = \bar{u} \tanh \left[\frac{Re}{2} (1-x) \right] \tag{11}$$

where

$$Re = 1/\mu \tag{12}$$

and $\bar{u}$ is the solution of the equation

$$\frac{\bar{u}-1}{\bar{u}+1} = \exp \{-\bar{u}Re\} \tag{13}$$

The slope of the steady state solution at the right end increases and that at the left end tends to zero as Re increases.

McCormack's method was used to integrate Equation (8) and three point central differences were used to calculate the metrics of the transformation. The stability limit for McCormack's method for this problem was determined using the empirical formula given by Tannehill et al. (9).

Results are presented for various values of Re in Figures 4-8. In all cases the steady state results using an adaptive grid and those obtained using an equispaced grid are compared with the exact solution. In Figure 4 results for Re = 1 are shown. The errors are very small (< 0.04%) in both cases but the peak error without an adaptive grid is about 1.82 times the peak error with an adaptive grid. In Figure 5 results are presented for Re = 2. The ratio of the peak errors is now about 4.90 and a significant improvement in accuracy is seen. However, in Figure 5, the adaptive grid shows a slightly larger error in the region $0 < x < 0.2$. This is due to the fact that the second point in the grid has moved to the right a substantial distance resulting in a higher error in this region.

Figure 6 presents results for Re = 3. The second point in this case moves so far to the right that the truncation error in calculating the transformation metrics in this region swamps the entire solution resulting in a solution that is worse than the one obtained using an equispaced grid. In order to prevent extreme stretching of the grid it is necessary to include a measure of the truncation error introduced in calculating the transformation metrics into the driving force g,

$$g = |x_\xi| - |x_\xi|_{av} + \theta \{ |u_\xi| - |u_\xi|_{av} \} \tag{14}$$

where θ is a constant. Since x_ξ is greater than zero and u_ξ is less than zero in this problem, Equation (14) can be written as

$$g = x_\xi - (x_\xi)_{av} - \theta \{u_\xi - (u_\xi)_{av}\} \tag{15}$$

Since the grid converges when g is a constant over the entire region, the transformation for the converged grid can be shown to be

$$\xi = 1 - fu - (1-f)(1-x) \qquad 0 \le f \le 1 \tag{16}$$

where f is a constant. Hence, an equivalent way of preventing extreme stretching is to define $\bar{u}$ as

$$\bar{u} = fu + (1-f)(1-x) \tag{17}$$

and the driving force g as

$$g = |\bar{u}_\xi| - |\bar{u}_\xi|_{av} \tag{18}$$

The error curve obtained for $Re = 3$ and $f = 0.7$ is also shown in Figure 6. A substantial decrease in error is seen, the ratio of the peak errors being about 3.80. Figures 7 and 8 present results for $Re = 5$ and $Re = 10$ respectively. In both cases a smoothed form of the solution as given by Equation (17) is used. The ratio of peak errors is about 2.23 for $Re = 5$ and 2.13 for $Re = 10$. Figure 9 shows the transformation obtained for the case $Re = 3$, $f = 0.7$. The uniform nature of the transformation is apparent.

A better measure of the total truncation error at a point (e) is

$$e \propto dx^2 \, u_{xxx} \tag{19}$$

which can be approximated in this case as

$$e \propto dx^2 \, u_x \tag{20}$$

which yields

$$e \propto u_\xi / \xi_x \tag{21}$$

Equation (21) suggests a driving force of the form

$$g = |u_\xi / \xi_x| - |u_\xi / \xi_x|_{av} \tag{22}$$

Results of using such a driving force for the case $Re = 3$ are presented in Figure 10. The errors obtained are comparable to the ones obtained using an optimal f. However, the advantage in using this new form of the driving force lies in eliminating the empiricism required in determining the optimal f. Similar results were obtained for all $Re < 5.0$. Excessive stretching was once again observed for higher values of Re, indicating the inaccuracy in estimating the error. The analysis and results presented in this and the preceding paragraph show that the method is limited only by the accuracy with which the total truncation error at a point can be estimated.

The second problem solved was the two-dimensional unsteady, linearized, viscous Burger's equation

$$u_t + u_x + u_y = \mu(u_{xx} + u_{yy}) \tag{23}$$

in a square domain with the initial conditions

$$u(x,0,0) = 1 + \frac{1 - \exp(Re(x-1))}{(1 - \exp(-Re))}$$

$$u(0,y,0) = 1 + \frac{1 - \exp(Re(y-1))}{(1 - \exp(-Re))} \qquad (24)$$

$$u = 1 \qquad \text{otherwise}$$

where

$$Re = 1/\mu \qquad (25)$$

and the boundary conditions

$$u(x,0,t) = 1 + \frac{1- \exp(Re(x-1))}{(1 - \exp(-Re))}$$

$$u(0,y,t) = 1 + \frac{1 - \exp(Re(y-1))}{(1 - \exp(-Re))} \qquad (26)$$

$$u(x,1,t) = 1$$

$$u(1,y,t) = 1$$

This problem has the steady state solution

$$u = 1 + \frac{(1 - \exp(Re(x-1)))\ (1 - \exp(Re(y-1)))}{(1 - \exp(-Re))^2} \qquad (27)$$

McCormack's method was used to integrate Equation (23) and three point central differences were used to calculate the metrics of the transformation. To prevent excessive stretching of the grid a smoothed version of the solution $(\bar{u})$

$$\bar{u} = fu + (1-f)\,(4-x-y)/2 \qquad 0 \leq f \leq 1 \tag{28}$$

is used to calculate the driving force.

Figure 11 shows the grid obtained for $Re = 5$ and $f = 0.3$. The error is calculated at the points shown in Figure 11 and a linear interpolation is used to calculate the error at the points corresponding to the equispaced grid. The results are presented in Figures 12-15, at each y station. The adaptive grid yields slightly higher errors in the low gradient region as in Figure 12 and gradually progresses to much lower errors in the high gradient regions as in Figure 15. The increases in accuracy are not as high as in the one-dimensional case, the main reason being the inaccuracy in establishing the local truncation error. One complication that exists only in two- and three-dimensional problems is the appearance of cross derivative terms in any estimate of the local truncation error. The absence of cross derivative terms in the present formulation of the grid generation scheme is felt particularly at the point $x = 0.8$, $y = 0.2$ in Figure 12. This point has a large value of u_x and a small value of u_y resulting in mesh clustering only in the x direction. However, the terms u_{xyy} and u_{xxy} are by no means small and hence due to large Δy in this region give rise to large errors. Future work with two-dimensional problems will require that the influence of cross derivative terms be included in the generation of grids.

TIME REQUIREMENTS

The number of integration steps required for convergence is always greater with an adaptive grid because of the lower values of maximum allowable time steps associated with mesh clustering. The ratio of the number of steps required with and without an adaptive grid goes all the way from 3.4 for $Re = 10$ to 1.4 for $Re = 1$ in the one-dimensional case and takes on a value of 2.3 in the two-dimensional case. However, time estimates will be given only on a per integration step basis. In the one-dimensional case the generation of the grid and recalculation of the transformation metrics takes less than 10% of the time taken for integration. In the two-dimensional case, the generation of the grid takes 25% and recalculation of metrics takes 70% of the time taken for integration. One of the reasons for the excessive time taken for the calculation of metrics is the presence of second derivatives like ξ_{xx}, ξ_{yy}, η_{xx}, and η_{yy}, all of which need to be determined numerically. The absence of these second derivatives greatly speeds up the calculation of metrics. Furthermore, if the problem requires the recalculation of metrics even without an adaptive grid, as in shock fitting programs, the time required to use an adaptive grid becomes very attractive. It must also be remembered that the percent extra time in this case is high because the equation being solved is very simple. Since the time for grid generation remains about the same in far more complicated problems, the present extra time for grid generation will be much less for such problems.

In conclusion, the major contributions of this paper are:

(a) Formulation of simple first order partial differential equations for the grid point velocity in transient problems.

(b) Significant error reductions for solutions of Burger's equation in one and two dimensions.

(c) The use of local flow information and boundary motion in determining the interior grid point motion.

REFERENCES

1. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: Automatic Numerical Generation of Body-Fitted Curvalinear Coordinate Systems for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. J. Comp. Phys., 15, 299, 1974.

2. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: TOMCAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies. J. Comp. Phys., vol. 24, no. 3, 1977.

3. Thames, F. C.: Numerical Solution of the Incompressible Navier-Stokes Equations about Arbitrary Two-Dimensional Bodies. Ph.D. Dissertation, Mississippi State Univ., Mississippi State, MS, 1975.

4. Middlecoff, J. F.; and Thomas, P. D.: Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations. AIAA Paper 79-1462, July 1979.

5. Hindman, R. G.; Kutler, P.; and Anderson, D. A.: A Two-Dimensional Unsteady Euler-Equation Solver for Flow Regions With Arbitrary Boundaries. AIAA Paper 79-1465, July 1979.

6. Dwyer, H. A.; Kee, R. J.; and Sanders, B. R.: An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer. AIAA Paper 79-1464, July 1979.

7. Olsen, J.: Subsonic and Transonic Flow Over Sharp and Round Nosed Nonlifting Airfoils. Ph.D. Dissertation, Ohio State University, Columbus, Ohio, 1976.

8. Pierson, B. L.; and Kutler, P.: Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics. AIAA Paper 79-0272, January 1979.

9. Tannehill, J. C.; Holst, T. L.; and Rakich, J. V.: Numerical Solution of Two-Dimensional Viscous Blunt Body Flows with an Impinging Shock. AIAA Paper 75-154, January 1975.

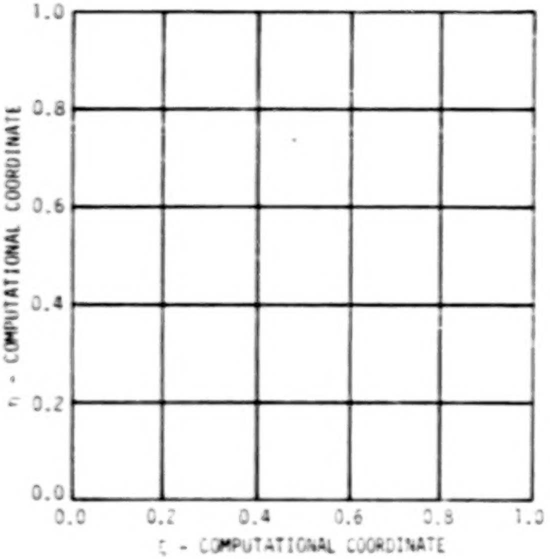

1.0
0.8
0.6
0.4
0.2
0.0
η - COMPUTATIONAL COORDINATE
0.0
0.2
0.4
0.6
0.8
1.0
ξ - COMPUTATIONAL COORDINATE

Fig. 1. Computational space.

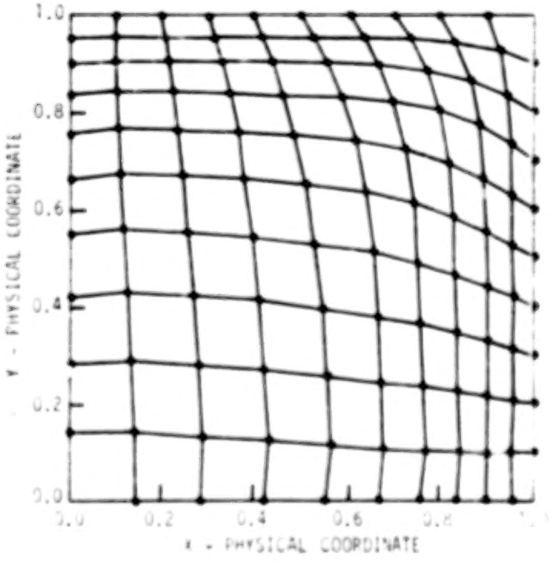

1.0
0.8
0.6
0.4
0.2
0.0
y - PHYSICAL COORDINATE
0.0
0.2
0.4
0.6
0.8
1.0
x - PHYSICAL COORDINATE

Fig. 2. Grid generated using aperiodic boundaries.

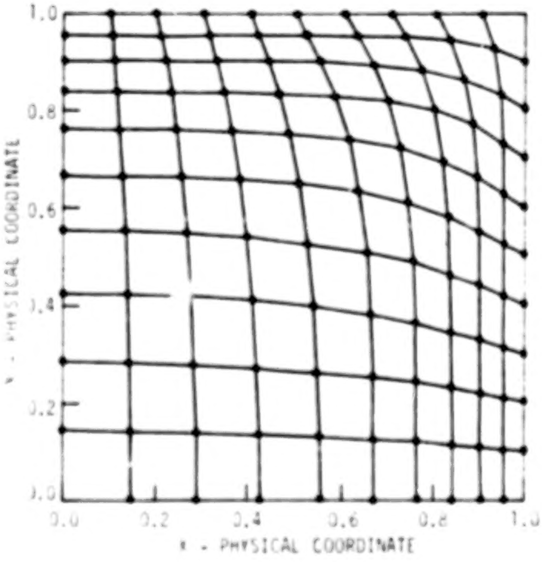

1.0
0.8
0.6
0.4
0.2
0.0
y - PHYSICAL COORDINATE
0.0
0.2
0.4
0.6
0.8
1.0
x - PHYSICAL COORDINATE

Fig. 3. Grid generated using periodic boundaries.

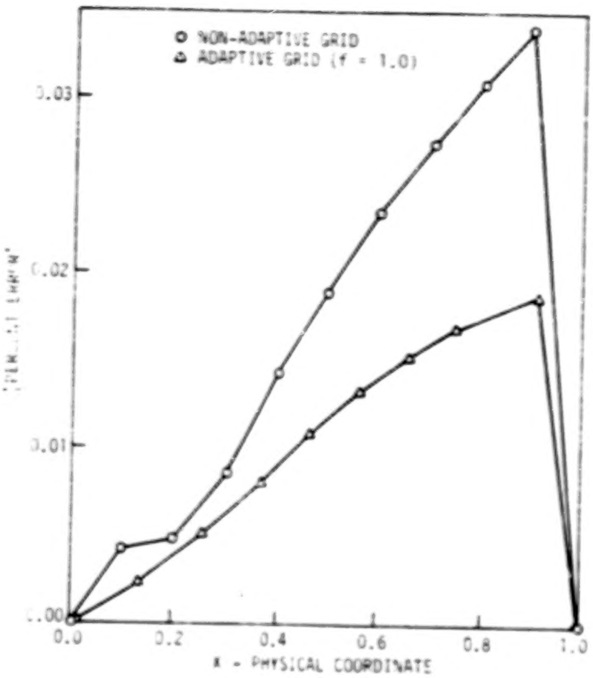


Fig. 4. Comparison of errors, Re = 1.0.

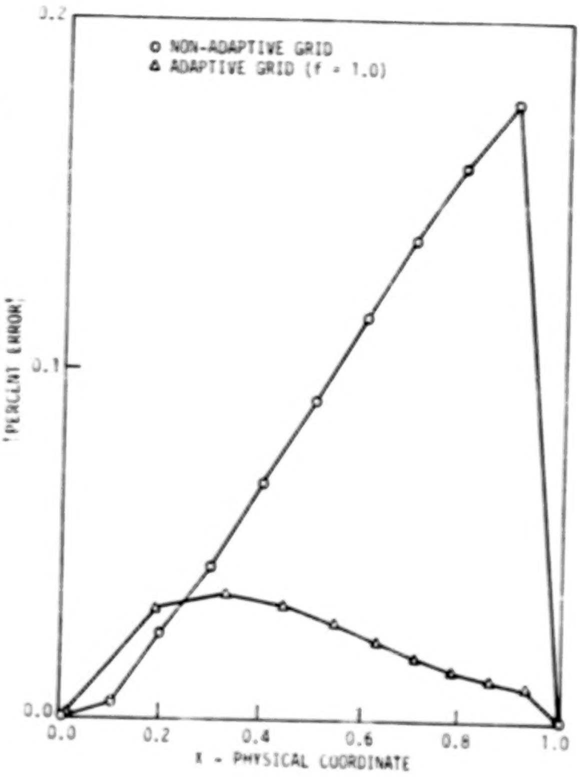


Fig. 5. Comparison of errors, Re = 2.0.

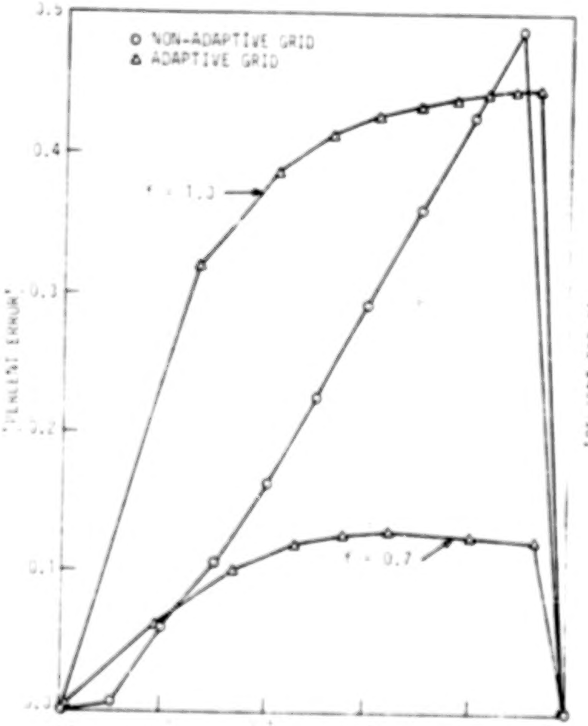


Fig. 6. Comparison of errors, Re = 3.0.

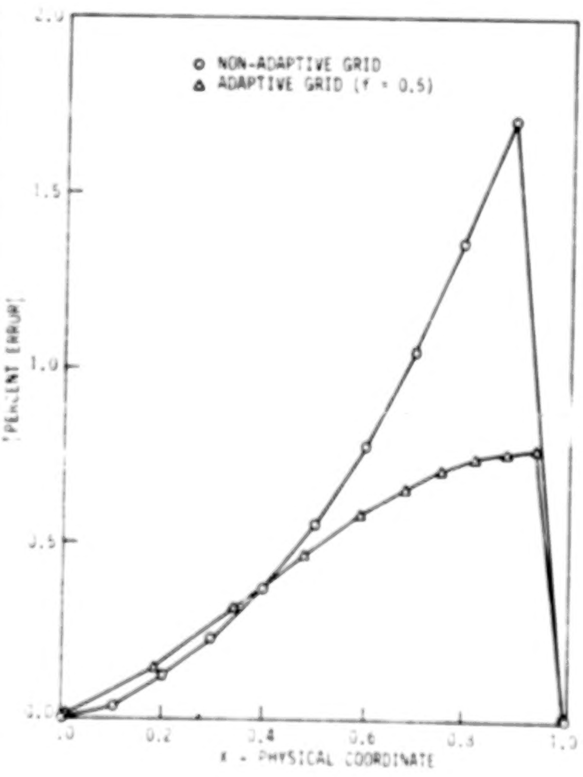


Fig. 7. Comparison of errors, Re = 5.0.

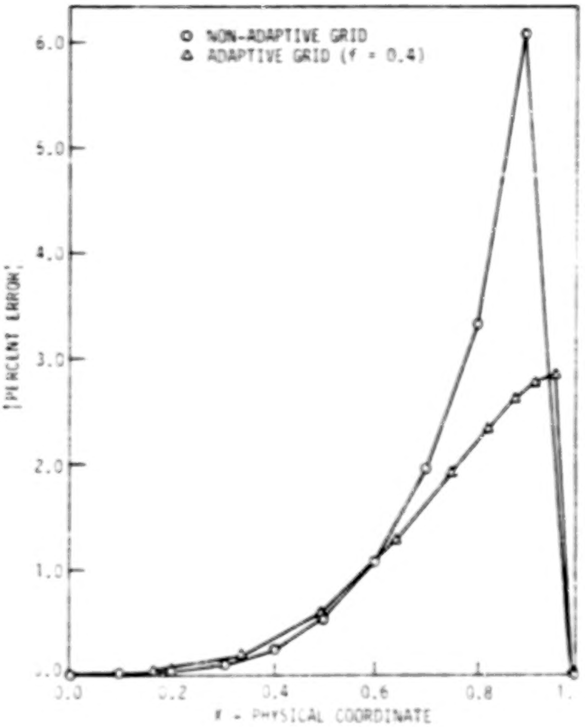


Fig. 8. Comparison of errors, Re = 10.0.

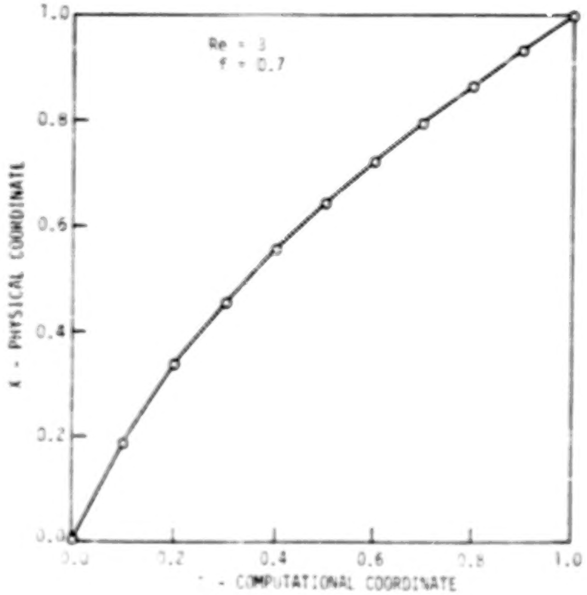


Fig. 9. Converged grid for the one-dimensional Burger's equation.

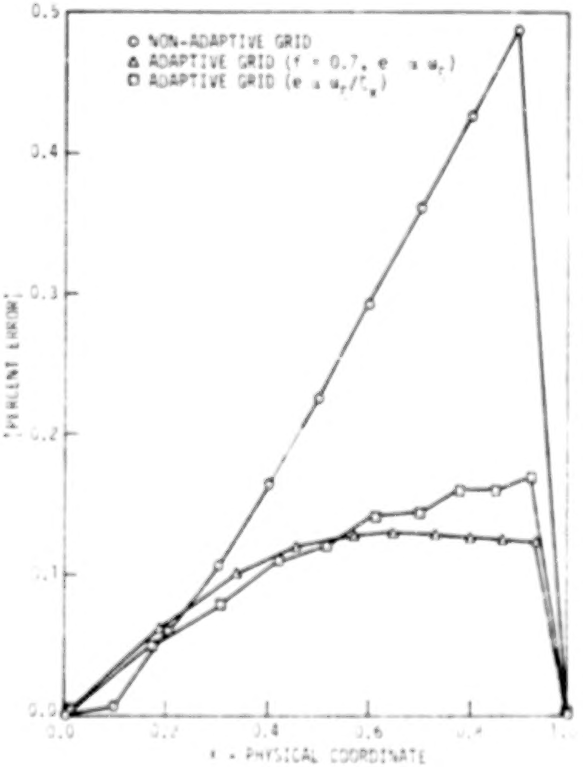


Fig. 10. Comparison of errors, Re = 3.0.

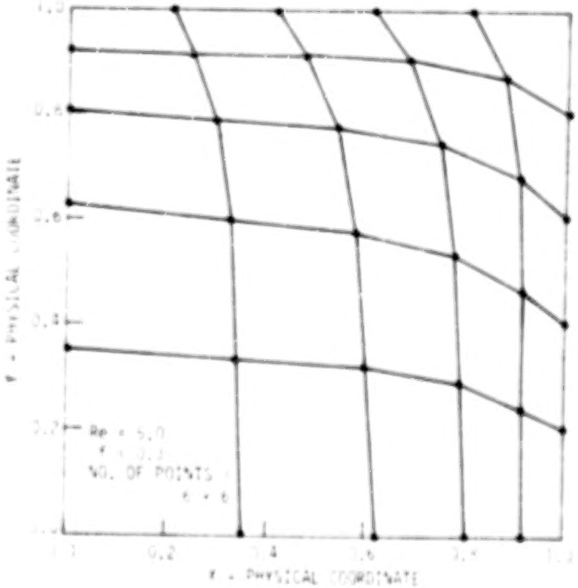

Re = 5.0
f = 0.3
NO. OF POINTS = 6 × 6
y - PHYSICAL COORDINATE
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x - PHYSICAL COORDINATE

Fig. 11. Converged grid for the two-dimensional Burger's equation.

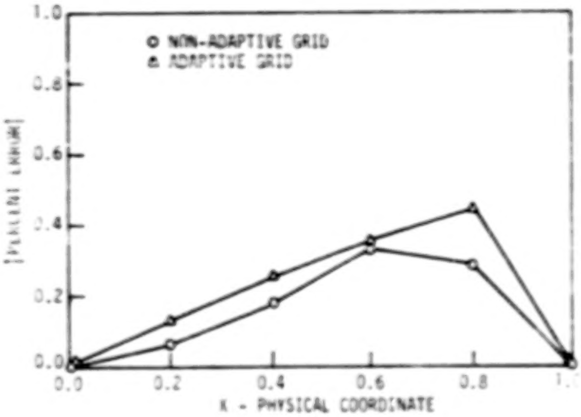


Fig. 12. Comparison of errors, Re = 5.0, Y = 0.2.

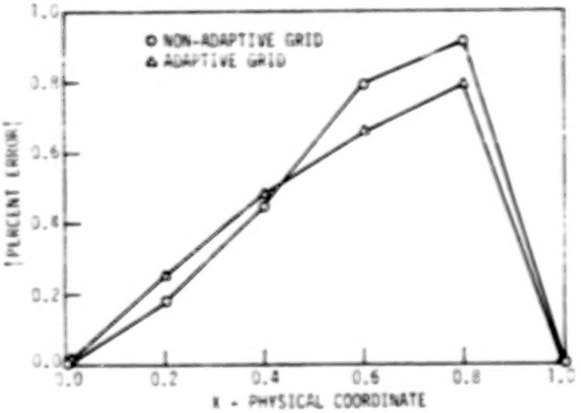


Fig. 13. Comparison of errors, Re = 5.0, Y = 0.4.

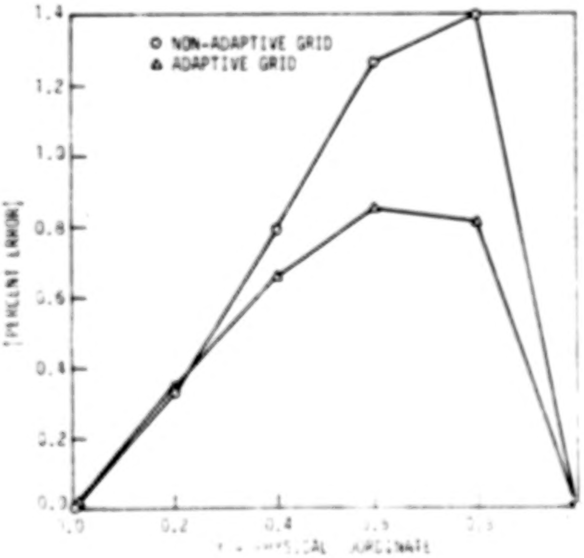


Fig. 14. Comparison of errors, Re = 5.0, Y = 0.6.

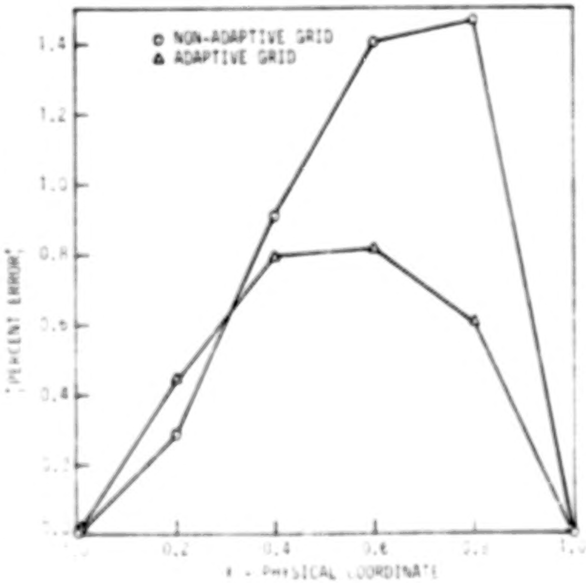


Fig. 15. Comparison of errors, Re = 5.0, Y = 0.8.

A Two Dimensional Mesh Verification Algorithm

R. Bruce Simpson
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada.

Abstract

A finite element mesh is commonly represented in a program by lists of data, i.e., vertex coordinates, element incidences, boundary data. In general, these lists describe a collection of triangles. Whether the triangles form proper mesh for a region or not, i.e. whether they 'tile' a region, is data dependent in a non obvious way. This paper specifies a set of conditions on the triangles (i.e. on the list data) which ensure that the triangles tile a region and which also can be verified by an algorithm which is referred to in the title and which is claimed to be of reasonable efficiency.

Basic List Representation of a Mesh

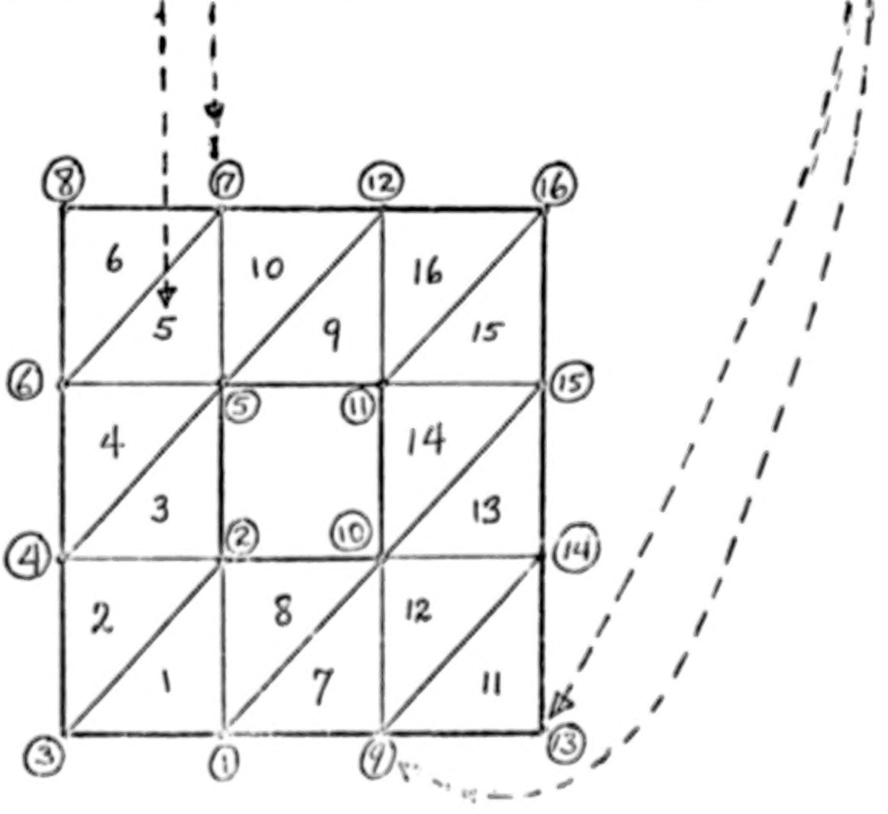
The mesh verification algorithm assumes that the collection of triangles is described by three lists as shown in the following small example.

Vertex Index	Coordinates x-y coordinates	
1	2.00	1.00
2	2.00	2.00
3	1.00	1.00
4	1.00	2.00
5	2.00	3.00
6	1.00	3.00
7	2.00	4.00
8	1.00	4.00
9	3.00	1.00
10	3.00	2.00
11	3.00	3.00
12	3.00	4.00
13	4.00	1.00
14	4.00	2.00
15	4.00	3.00
16	4.00	4.00

Element incidences index	vertex indices		
1	1	2	3
2	2	4	3
3	2	5	4
4	5	6	4
5	5	7	6
6	7	8	6
7	9	10	1
8	10	2	1
9	11	12	5
10	12	7	5
11	13	14	9
12	14	10	9
13	14	15	10
14	15	11	10
15	15	16	11
16	16	12	11

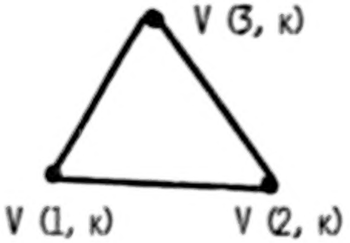
Boundary References index	references	
1	1	3
2	2	2
3	3	1
4	4	2
5	6	1
6	6	2
7	7	3
8	8	1
9	9	3
10	10	1
11	11	1
12	11	3
13	13	1
14	14	2
15	15	1
16	16	1

indicates a boundary edge starts at ⟶ - - →



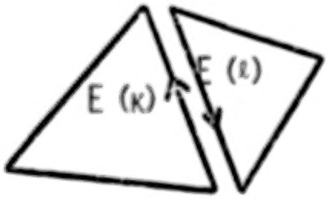
CONDITIONS

C 1) THE TRIANGLE VERTICES ARE SPECIFIED IN COUNTER CLOCKWISE ORDER



C 2) EITHER THE ITH EDGE OF E (K) IS THE ONLY EDGE JOINING ITS END POINTS (BOUNDARY ELEMENT)

OR THERE IS EXACTLY ONE ELEMENT, E (ℓ) HAVING THE SAME EDGE. IN THIS LATTER CASE, THE DIRECTIONS OF THIS LINE SEGMENT AS EDGES OF E (K) AND E (ℓ) MUST BE DIFFERENT.



C 3) NO BOUNDARY EDGE INTERSECTS MORE THAN ONE ELEMENT, EXCEPT AT ITS END POINTS.

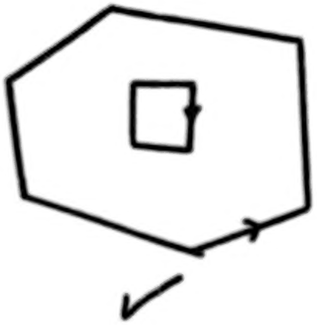
C 4) A VERTEX CAN HAVE AT MOST ONE BOUNDARY EDGE DIRECTED AWAY FROM IT.

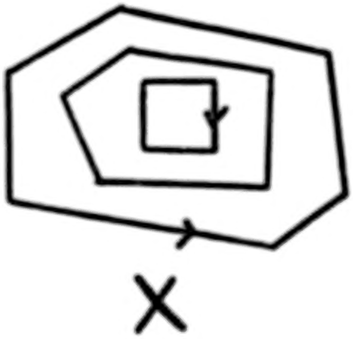
IMPLICATIONS

1) Mesh boundary edges form a set of disjoint, oriented, simple closed curves

$$C_1, \; C_2, \; \ldots, \; C_K \quad \equiv \quad \text{mesh boundary curves}$$

2) Each curve of bounded interior defines a connected region. The boundary of this region is composed of mesh boundary curves





(Assume 1 curve of bounded interior - C_1)

$$\text{Define} \quad R \equiv \bigcap_{i=1}^{K} \quad (\text{interior of } C_i)$$

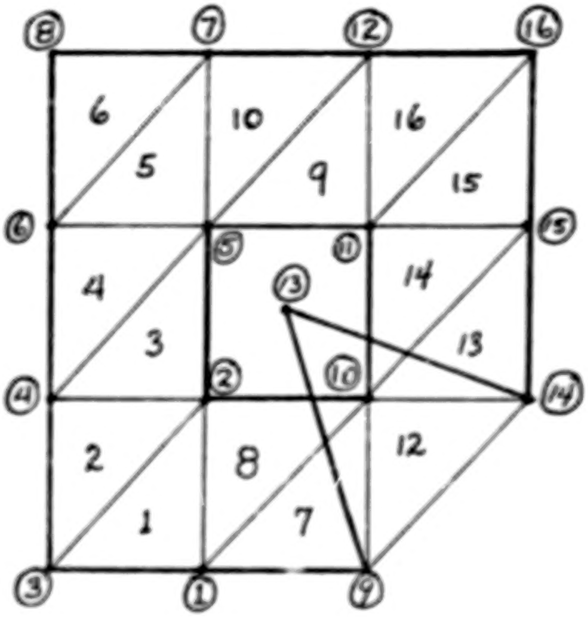
(connectivity K)

3) $R = \bigcup_{j=1}^{N_E} E(j)$

4) If $P \in R$, P is not an element edge

$\Longrightarrow$ P lies in exactly one element.

Small Example Invalid Mesh on Hollow Square



Coordinates of vertex 13 changed to (2.5, 2.5)

Section of Mesh Verification Algorithm Detailed Error Report

```
 MESH VERIFICATION ERROR

    INTERSECTING BOUNDARY EDGES -
    EDGE FROM VERTEX 13 AT ( 2.50, 2.50)   TO VERTEX  9 AT ( 3.00, 1.00)
    EDGE FROM VERTEX  2 AT ( 2.00, 2.00)   TO VERTEX 10 AT ( 3.00, 2.00)

 MESH VERIFICATION ERROR

    INTERSECTING BOUNDARY EDGES -
    EDGE FROM VERTEX 14 AT ( 4.00, 2.00)   TO VERTEX 13 AT ( 2.50, 2.50)
    EDGE FROM VERTEX 10 AT ( 3.00, 2.00)   TO VERTEX 11 AT ( 3.00, 3.00)

  FROM EDGCAN, NO. OF BOUNDARY CURVES=     2

 MESH VERIFICATION ERROR

    ELEMENT  11 APPEARS TO HAVE VERTICES LISTED IN WRONG ORDER
            X=     3.000000E 00    Y=     1.000000E 00
            X=     2.500000E 00    Y=     2.500000E 00
            X=     4.000000E 00    Y=     2.000000E 00
            DET =    -2.000000E 00

MESH CHECK ENCOUNTERED       3 ERRORS
```

BLANK PAGE

BLANK PAGE

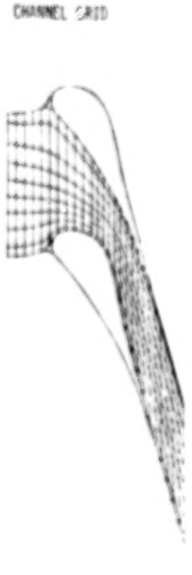
GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION

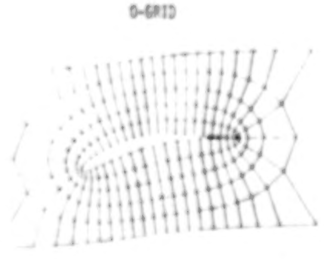
Peter M. Sockol
NASA Lewis Research Center
Cleveland, Ohio

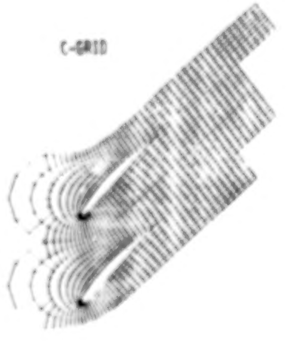
ABSTRACT

This paper presents a rapid procedure for generating C-type cascade grids suitable for viscous flow computations in turbomachinery blade rows. The resulting mesh is periodic from one blade passage to the next, nearly orthogonal, and continuous across the wake downstream of a blade. The procedure employs a pair of conformal mappings that take the exterior of the cascade into the interior of an infinite strip with curved boundaries. The final transformation to a rectangular computational domain is accomplished numerically. The boundary values are obtained from a panel solution of an integral equation and the interior values by a rapid ADI solution of Laplace's equation. Examples of C-type grids are presented for both compressor and turbine blades and the extension of the procedure to three dimensions is briefly outlined.

Most of the coordinate systems in current use for turbomachinery flow computations are of one of three types. The channel grid has one family of lines starting upstream, passing through the blade rows, and continuing on downstream. The O-type grid has one family of lines that form closed loops around the blades. Finally, the C-type grid has one family of lines that wrap around the blade leading edge and continue on downstream. While the channel grid can be aligned with the flow and is fairly easy to generate, the resolution around the leading edge is usually poor and a choice usually has to be made between periodicity or near orthogonality for highly staggered cascades. Although the O-type grid provides excellent resolution around leading and trailing edges and may be both periodic and orthogonal, in general there is no mesh line aligned with the downstream flow and, hence, it is unsuitable for viscous computation. The C-type grid, on the other hand, appears to be a good choice for viscous flows. It provides good leading edge resolution, it can be both periodic and orthogonal, and it can be aligned with the downstream flow. This paper presents a rapid procedure for generating such C-type grids.

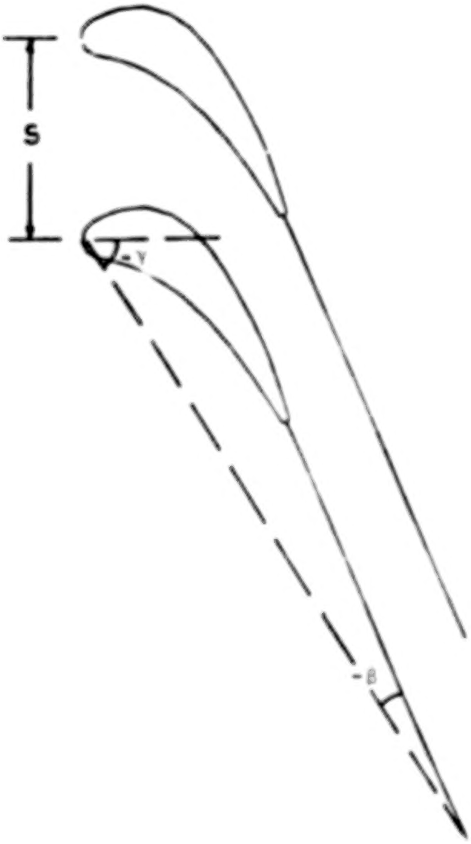






The procedure starts with a conformal mapping which takes the exterior of a cascade of semi-infinite flat plates in the Z-plane into the interior of the unit circle in the W-plane. Upstream infinity maps to the origin and downstream infinity to +1 on the real axis. This mapping is a limiting form of the standard mapping for a cascade of finite-chord flat plates (1). When this mapping is applied to a real geometry, such as the turbine cascade in the figure, the semi-infinite flat plate is taken to run from a point Z_1 inside the leading edge through the downstream end of the wake.

CONTOUR IN CASCADE PLANE

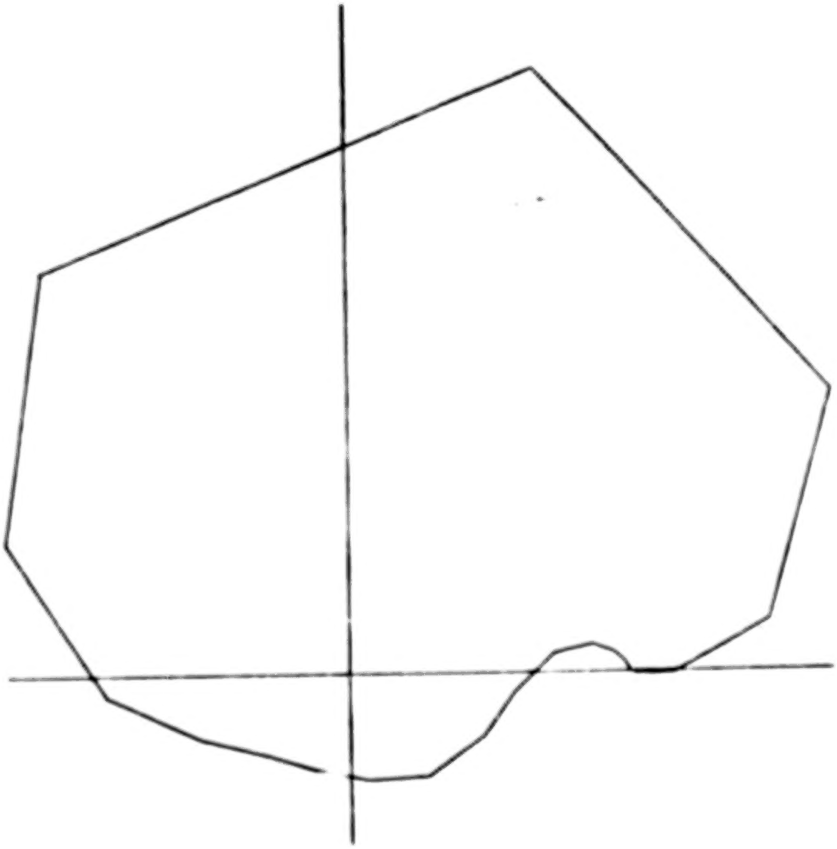


$$Z = Z_0 + \lambda\left[e^{-i\gamma}(\log W - i\pi) - 2\cos\gamma \log(1 - W)\right]$$

$$\lambda = \frac{S}{2\pi} e^{i\gamma}, \quad Z_0 = Z_1 + 2\lambda\left[\gamma \sin\gamma + \cos\gamma \log(2\cos\gamma)\right]$$

The mapping of the turbine cascade and wakes of the preceding figure produces the highly distorted "circle" in the adjacent figure. Note that the contour actually crosses the real axis twice between zero and one. The next mapping takes the interior of the unit circle in the W plane, with a branch cut from zero to one along the real axis, to the interior of the infinite strip between the real axis and $(-i\frac{\pi}{2})$ in the ζ-plane. The upper and lower sides of the wake at downstream infinity are mapped to plus and minus infinity, respectively, while upstream infinity maps to the origin. Since W is a function of ζ^2, reflection of ζ through the origin leaves W unchanged.

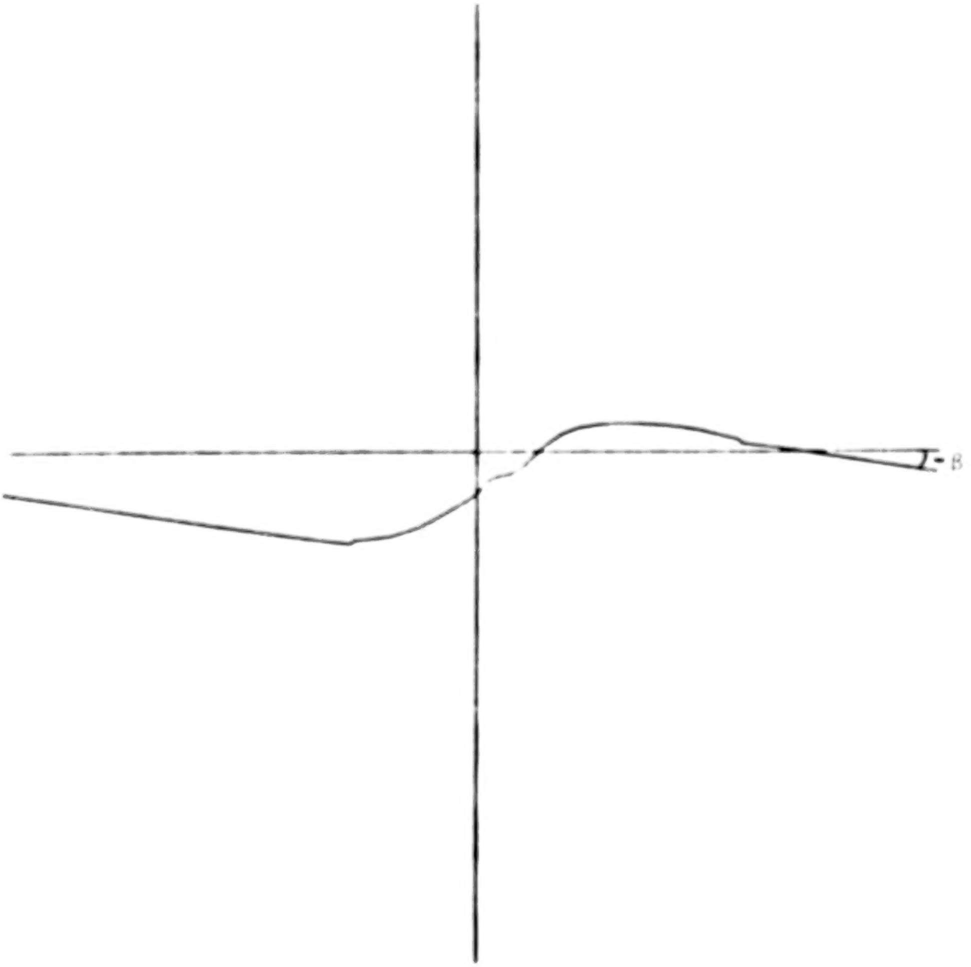
CONTOUR IN W PLANE



$W = \tanh^2 \frac{\zeta}{2}$

The image of the cascade of turbine blades and wakes in the ζ -plane is a pair of parallel straight lines connected by a roughly S-shaped curve. In actual practice W is eliminated between the two functions and the transformation from Z to ζ is obtained by Newton iteration proceeding from point to point around the contour. To insure that the branch cuts of the logarithms are never crossed, the imaginary parts of these logarithms are saved. Whenever the change in either of these quantities between adjacent points exceeds $\pm\pi$, the computed value of the logarithm at the new point is incremented by $\mp 2\pi i$, i.e. in the opposite direction.

CONTOUR IN ζ PLANE



The final mapping transforms the infinite strip in the ζ -plane, bounded by the blade-wake contour and its reflection through the origin, into a rectangular domain with coordinates $F = \xi + i\eta$. If we let F be the complex potential for flow through the strip and require $F(\zeta) = -F(-\zeta)$ together with $\eta = -1$ along the contour, then we can write F as a contour integral. Here C, β, and h are, respectively, the complex velocity, flow angle, and normal channel width in the far field. The figure shows the formation of an integral equation for the unknown vortex density q_t. The source density q_n is set to cancel the normal component of the velocity C. Here s is distance along contour. A simple panel method, with flat panels and locally constant q_t and q_n, is used to solve for q_t and then to find ξ as a function of ζ along the contour.

FORMATION OF INTEGRAL EQUATION

COMPLEX POTENTIAL

$$F(\zeta) = \frac{1}{2\pi i}\int_{-\infty}^{\infty} \gamma(t)\,\log\frac{t+\zeta}{t-\zeta}\,dt + [illegible]$$

WITH F [illegible] $+ i$ [illegible] $\frac{2}{h}e^{-i\beta}$

SET $\gamma(t)\frac{dt}{ds} = q_t - iq_n$

WITH $q_n(t) = \mathrm{Im}\left[C\frac{dt}{ds}\right]$

ON CONTOUR $\eta = -1$ AND

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\left[q_t\,\mathrm{Re}\left\{\log\frac{t+\zeta}{t-\zeta}\right\} + q_n\,\mathrm{Im}\left\{\log\frac{t+\zeta}{t-\zeta}\right\}\right]ds$$

$$= 1 + \mathrm{Im}\ [illegible]$$

$$[illegible]\ \mathrm{Re}\ [illegible] = -\frac{1}{2\pi}\int_{-\infty}^{\infty}\left[q_n\,\mathrm{Re}\left\{\log\frac{t+\zeta}{t-\zeta}\right\}\right.$$

$$\left. - q_t\,\mathrm{Im}\left\{\log\frac{t+\zeta}{t-\zeta}\right\}\right]ds$$

Generation of the grid in the rectangular ξ , η space proceeds in two stages. First points are located on the boundaries such that the grid in the cascade plane is periodic and continuous across the wake. Periodicity is enforced by distributing points symmetrically about the origin along the ξ -axis. Continuity across the wake, away from the trailing edge, is achieved by selecting a constant mesh size $\Delta\xi$ for this region such that the spacing in the cascade plane is an integer fraction of the staggered distance $s \sin |\beta_w|$, where s is the blade pitch and β_w is the wake angle. The values of ζ along the boundary are then found by inverse interpolation in the solution of ξ vs ζ . In order to enforce continuity near the trailing edge, a local straining is first introduced that places a point directly at the trailing edge. Then pairs of neighboring points across the wake are adjusted until their images in the cascade plane coincide. The distribution of points with η at the two ends of the region is arbitrary. Uniform spacing can be used for inviscid flows, and boundary layer stretching can be used to cluster points near the blade surface and wake for viscous flows. Once the boundary values of ζ are specified, interior values are found by solving the complex Laplace equation by a cyclic ADI relaxation scheme which has the symmetry properties of ζ built in. Estimates of the maximum and minimum eigenvalues of the matrix are used to obtain near optimum values of the acceleration parameters (2). Convergence to the round-off error limit with seven place arithmetic is typically obtained in six to twelve iterations, even for cases where the maximum and minimum eigenvalues differ by five orders of magnitude.

GRID GENERATION

- BOUNDARY VALUES IN (ξ , η) SPACE

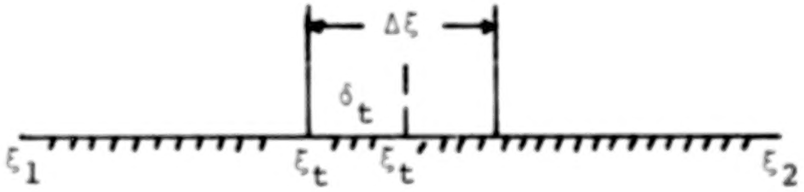
  ξ VALUES SYMMETRIC ABOUT ORIGIN

  $\Delta\xi$ CONSTANT ALONG WAKE WITH

  $$|\Delta z| = \left(s \sin |\beta_w|\right) / \text{integer}$$

  LOCAL STRAINING NEAR TRAILING EDGE

  $$\xi' = \xi + \delta_t \left[\frac{\xi - \xi_1}{\xi_t - \xi_1} \cdot \frac{\xi - \xi_2}{\xi_t - \xi_2}\right]^2$$



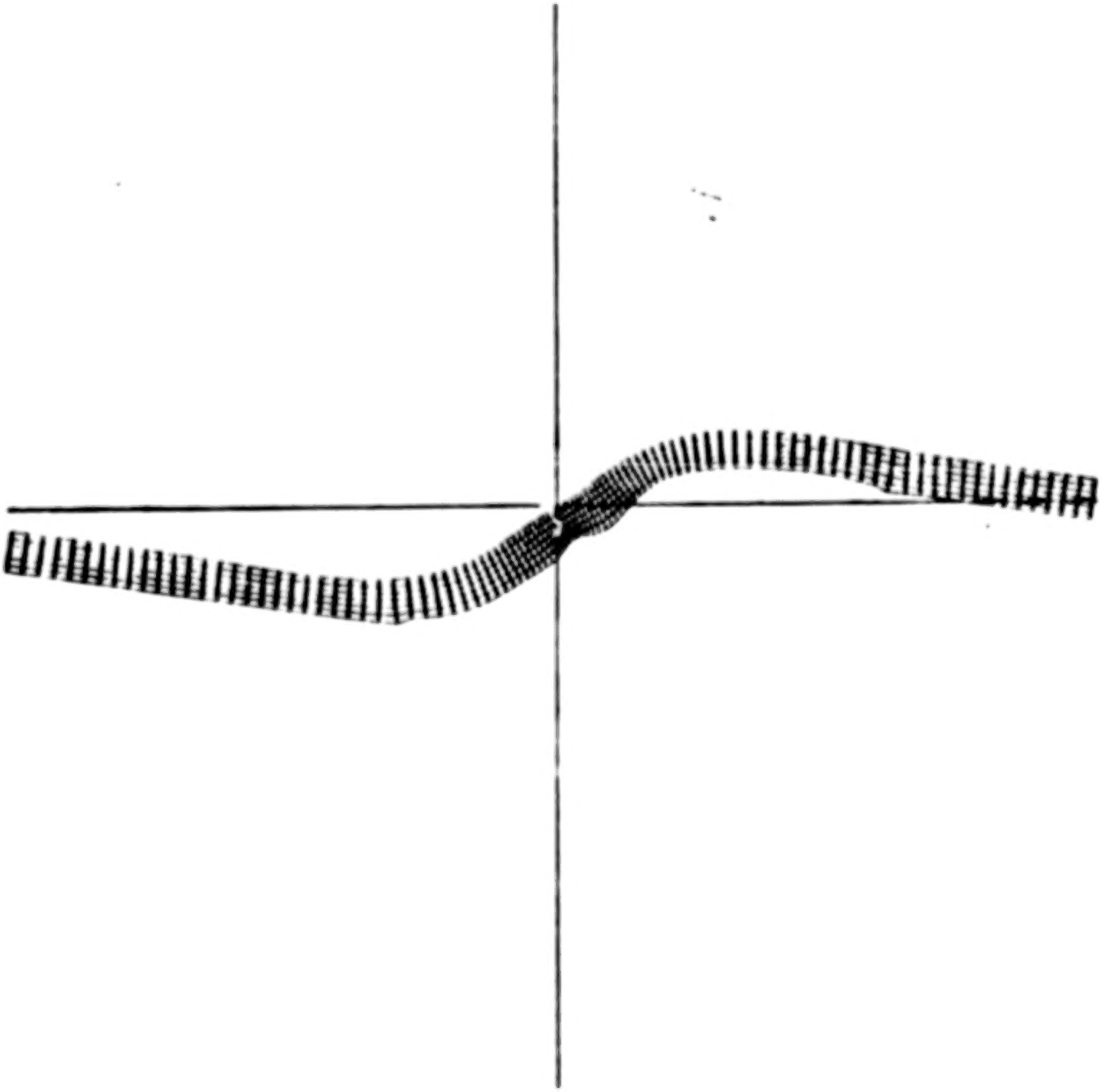
  ARBITRARY DISTRIBUTION ALONG η

- INTERIOR VALUES IN (ξ , η) SPACE

  $$\frac{\partial^2 \zeta}{\partial \xi^2} + \frac{\partial^2 \zeta}{\partial \eta^2} = 0$$

  SOLVED BY CYCLIC ADI RELAXATION

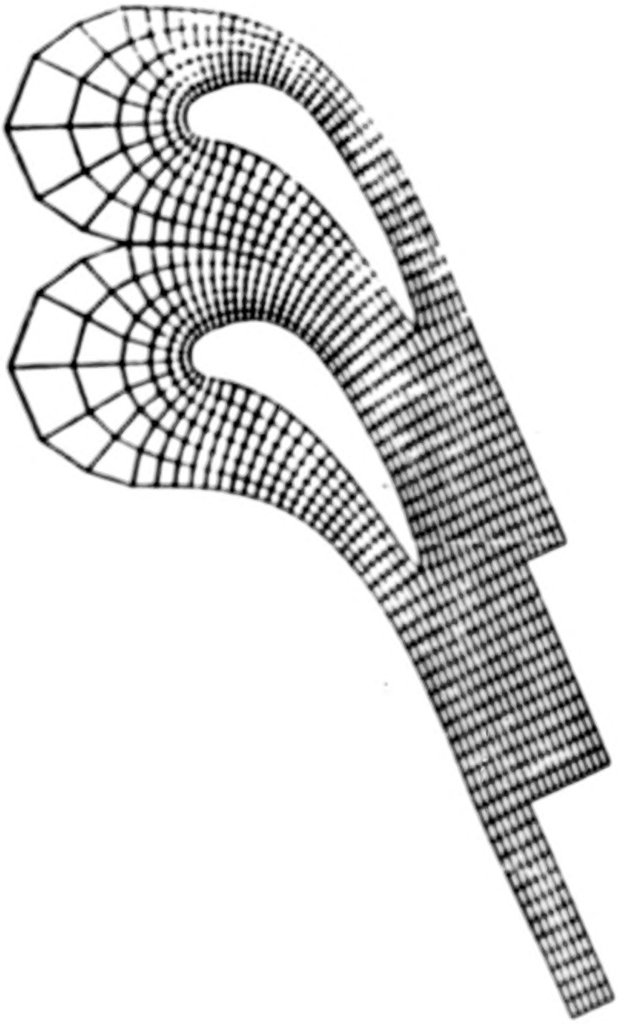
This figure shows the grid distribution in the ζ -plane. Note that the upper boundary in the plot, which is found by the symmetric ADI solution of Laplace's equation, maps into the upper and lower periodic lines in the cascade plane.



GRID IN ζ PLANE

The final grid in the cascade plane is obtained by conformal mapping of the solution in the ζ -plane using the two analytic functions previously introduced. This figure shows the grid distribution for the cascade of turbine blades. Note that the continuity across the wake was obtained at the expense of a small amount of nonorthogonality. The rounded cap at the upstream boundary was obtained by extrapolation from the next two inner loops. Generation of this grid (99 x 7 points) required about 1.4 sec. of CPU time on an IBM 3033 computer.

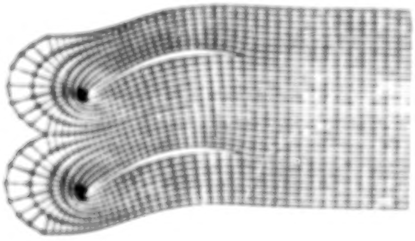
GRID IN CASCADE PLANE



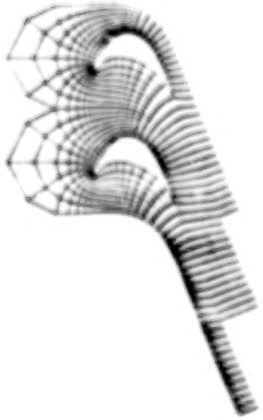
The last figure presents C grids for a compressor stator and a turbine rotor. The stator was designed to turn the flow to the axial direction, hence, there is zero stagger in the downstream boundary. The turbine rotor is a particularly difficult case as it was designed to produce 126 degrees of turning. In this case the imposition of continuity across the wake resulted in a significant change in slope.

The extension of this procedure to the generation of three-dimensional turbomachinery grids should be relatively straightforward. First the spanwise direction is discretized by a number of coaxial, axisymmetric surfaces. Next, and most difficult, the intersection of the blade with each of these surfaces is obtained in meridional (m) and tangential (θ) coordinates. Since the geometry is periodic in θ , these (m , θ) coordinates can be fed into the current program to generate a C-grid on each of the axi-symmetric surfaces. For O-grids this has already been done by Dulikravich (3).

COMPRESSOR STATOR



TURBINE ROTOR



REFERENCES

1. Kober, H.: Dictionary of Conformal Representations, Dover Publications, Inc., Second ed. with corrections, 1957, p. 131.

2. Varga, R. S.: Matrix Iterative Analysis, Prentice-Hall, Inc., 1962.

3. Dulikravich, D. S.: Fast Generation of Body Conforming Grids for 3-D Axial Turbomachinery Flow Calculations. Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 16 of this compilation.)

NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES

Reese L. Sorenson
NASA Ames Research Center, Moffett Field, CA 94035

and

Joseph L. Steger*
Flow Simulations, Inc., Sunnyvale, CA 94086

Abstract

A new method for generating boundary-fitted, curvilinear, two-dimensional grids by the use of the Poisson equations is presented. Grids of C-type and O-type have been made about airfoils and other shapes, with circular, rectangular, cascade-type, and other outer boundary shapes. Both viscous and inviscid spacings have been used. In all cases two important types of grid control can be exercised at both inner and outer boundaries. First is arbitrary control of the distances between the boundaries and the adjacent lines of the same coordinate family, i.e., "stand-off" distances. Second is arbitrary control of the angles with which lines of the opposite coordinate family intersect the boundaries. Thus, both grid cell size (or aspect ratio) and grid cell skewness are controlled at boundaries. Reasonable cell size and shape are ensured even in cases wherein extreme boundary shapes would tend to cause skewness or poorly controlled grid spacing. An inherent feature of the Poisson equations is that lines in the interior of the grid smoothly connect the boundary points (the grid mapping functions are second-order differentiable).

A user-oriented, well documented, FORTRAN computer program, called GRAPE, has been written to employ this grid generation method. It is available from the Applied Computational Aerodynamics Branch at NASA-Ames Research Center.

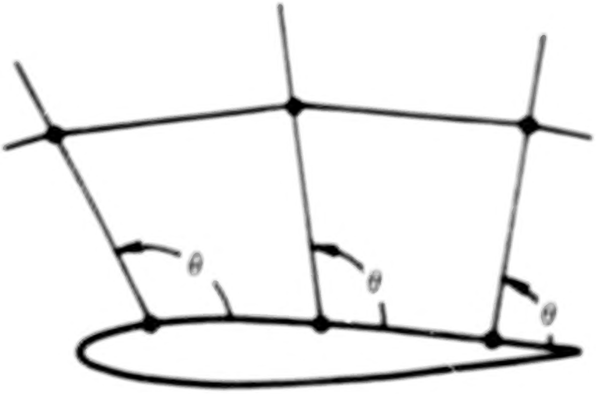
---

*Now with Stanford University.

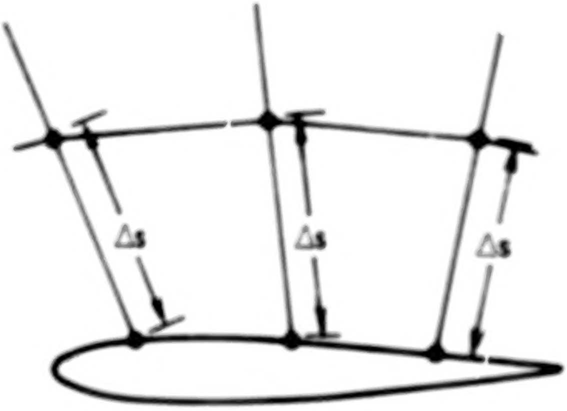
DESIRED PROPERTIES OF A GRID GENERATOR

- ARBITRARY BOUNDARY SHAPES
- ARBITRARY POINT DISTRIBUTION ON BOUNDARIES
- SMOOTH VARIATION (DIFFERENTIABILITY) IN INTERIOR
- COMPUTATIONALLY FAST
- EASY TO USE

- CONTROL OF ANGLES AT BOUNDARIES

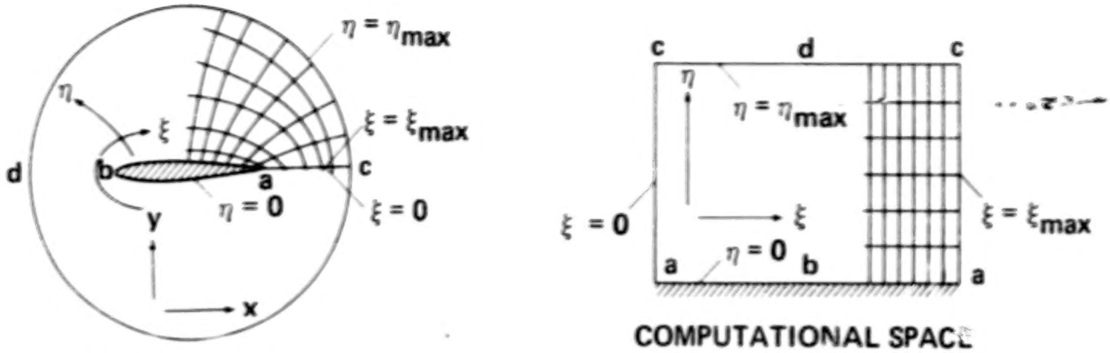


- CONTROL OF SPACING NEAR BOUNDARIES



The principal contribution of this work is that the angles and spacing at the boundaries are input. Thus, one need not try to implement pre-determined angles and spacing by trial-and-error tuning of other parameters.

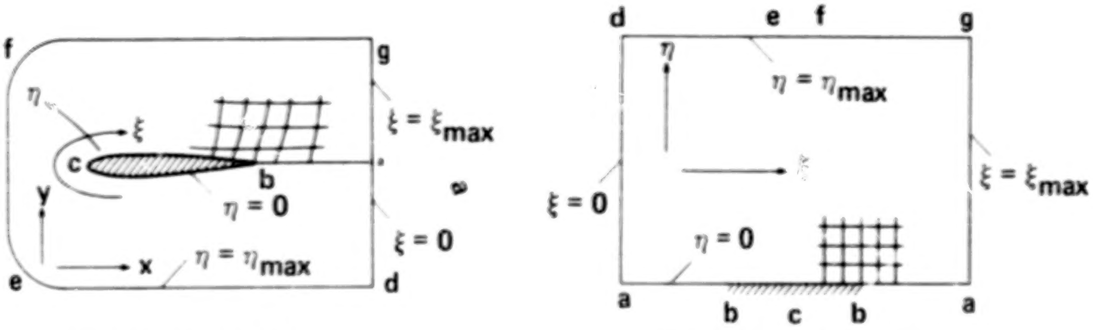
TOPOLOGY OF GRID MAPPINGS


η = η_max
η
ξ
ξ = ξ_max
d
b
a
c
η = 0
ξ = 0
y
x
c
d
c
η
η = η_max
ξ = 0
ξ
ξ = ξ_max
a
η = 0
b
a

PHYSICAL SPACE

COMPUTATIONAL SPACE

O-TYPE GRIDS


f
g
η
ξ
ξ = ξ_max
c
b
η = 0
ξ = 0
y
x
η = η_max
e
d
d
e
f
g
η
η = η_max
ξ
ξ = 0
ξ = ξ_max
η = 0
a
b
c
b
a

PHYSICAL SPACE

COMPUTATIONAL SPACE

C-TYPE GRIDS

Topology and notation for O-type and C-type grids are shown here. The independent variables in the physical space are x and y, while ξ and η are the independent variables in the cartesian computational space.

POISSON EQUATIONS

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta)$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta)$$

OR, WITH DEPENDENT AND INDEPENDENT VARIABLES INTERCHANGED:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2 (P x_\xi + Q x_\eta)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2 (P y_\xi + Q y_\eta)$$

WHERE: $\alpha = x_\eta^{\,2} + y_\eta^{\,2}$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^{\,2} + y_\xi^{\,2}$$

$$J = x_\xi y_\eta - x_\eta y_\xi$$

Basic to the method is that the grid transformations $\xi = \xi(x,y)$, $\eta = \eta(x,y)$ must satisfy the Poisson equations. The equations are solved with dependent and independent variables interchanged to facilitate numerical integration and the application of boundary conditions. The equations with variables thusly interchanged are sometimes referred to as the "transformed" Poisson equations.

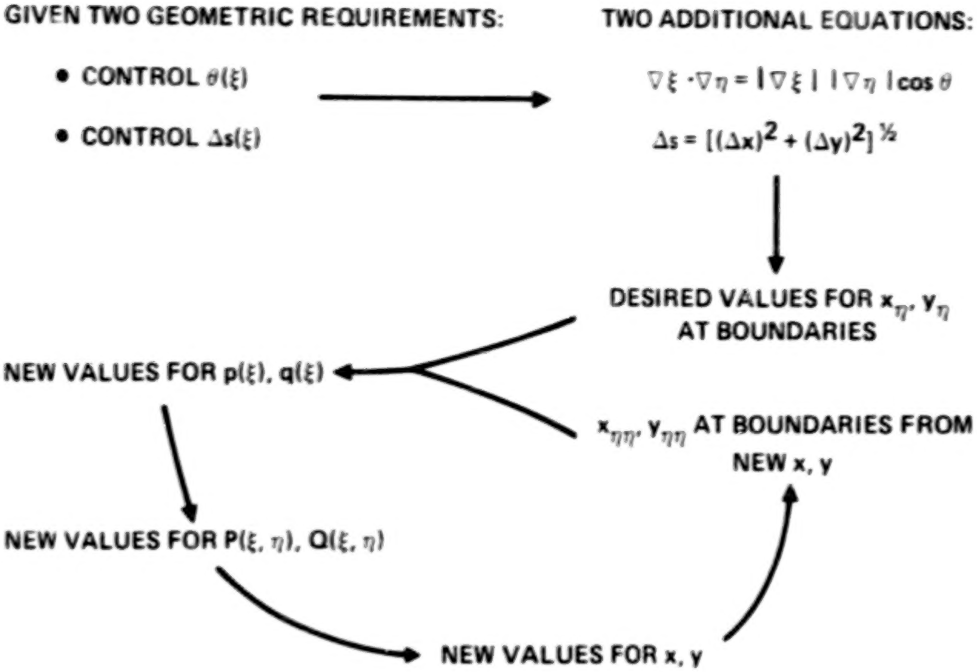
CHOICE OF INHOMOGENEOUS TERMS

$$P(\xi, \eta) = p(\xi)e^{-a\eta} + r(\xi)\, e^{-c(\eta_{max}-\eta)}$$

$$Q(\xi, \eta) = q(\xi)\, e^{-b\eta} + s(\xi)\, e^{-d(\eta_{max}-\eta)}$$

Inhomogeneous, or right-hand-side, or P and Q terms in the Poisson equations determine the character of the grid. Different choices for P and Q produce different grids. In this method P and Q are chosen as shown here with a,b,c, and d positive Note that the inner ($\eta = 0$) boundary $P(\xi,\eta)$ reduces to $p(\xi)$, and that at the outer ($\eta = \eta_{max}$) boundary $P(\xi,\eta)$ becomes $r(\xi)$, and similarly for $Q(\xi,\eta)$. The approach is to assume that the geometric input requirements (control of angles and spacing at boundaries) are satisfied along with the Poisson equations at the boundaries, then back-solve for $p(\xi)$, $q(\xi)$, $r(\xi)$, and $s(\xi)$. Then $P(\xi,\eta)$ and $Q(\xi,\eta)$ can be calculated for every point in the field.

ITERATIVE UPDATE OF INHOMOGENEOUS TERMS



GIVEN TWO GEOMETRIC REQUIREMENTS:

- **CONTROL $\theta(\xi)$**
- **CONTROL $\Delta s(\xi)$**

TWO ADDITIONAL EQUATIONS:

$$\nabla\xi \cdot \nabla\eta = |\nabla\xi|\,|\nabla\eta| \cos\theta$$

$$\Delta s = [(\Delta x)^2 + (\Delta y)^2]^{1/2}$$

DESIRED VALUES FOR x_η, y_η AT BOUNDARIES

NEW VALUES FOR $p(\xi)$, $q(\xi)$

$x_{\eta\eta}$, $y_{\eta\eta}$ AT BOUNDARIES FROM NEW x, y

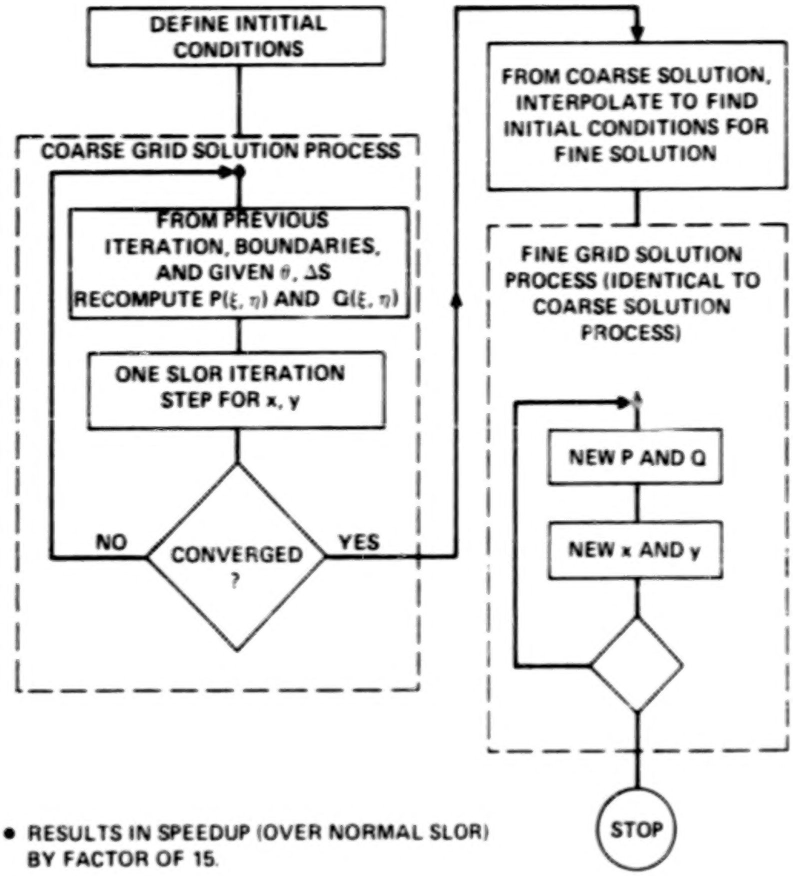
NEW VALUES FOR $P(\xi, \eta)$, $Q(\xi, \eta)$

NEW VALUES FOR x, y

For each boundary (inner and outer) the two geometric control requirements can be re-cast as the two additional equations shown on the upper right. These two equations can be solved for the two derivatives x_η and y_η. Derivatives x_ξ, y_ξ, $x_{\xi\xi}$, and $y_{\xi\xi}$ can be found by differencing known, fixed boundary data. Derivatives $x_{\xi\eta}$ and $y_{\xi\eta}$ are found by differencing the x_η and y_η, just found, with respect to ξ. Thus, to back-solve the transformed Poisson equations for $p(\xi)$ and $q(\xi)$, two derivatives remain to be found: $x_{\eta\eta}$ and $y_{\eta\eta}$.

In the solution procedure, each iteration step is in two parts. First, the x and y from the previous iteration (or initial conditions) are differenced to find $x_{\eta\eta}$ and $y_{\eta\eta}$ at the boundaries. These are combined with all the other derivatives discussed above, which are fixed for all iteration levels, to form the transformed Poisson equations at the boundary. These are back-solved for new values of $p(\xi)$ and $q(\xi)$. Terms $r(\xi)$ and $s(\xi)$ are similarly found. New values for $P(\xi,\eta)$ and $Q(\xi,\eta)$ can then be calculated. The second part of each solution step is to perform one iteration of some solution procedure, such as SLOR. The above is iterated to convergence, producing a grid that satisfies the given geometric requirements. Inhomogeneous terms which yield the desired grid control are thus found automatically as the solution proceeds.

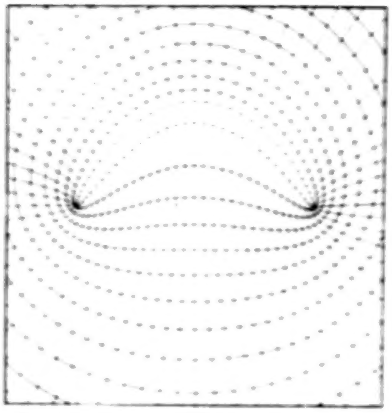
COARSE-FINE SEQUENCING SOLUTION PROCEDURE


DEFINE INTITIAL CONDITIONS
COARSE GRID SOLUTION PROCESS
FROM PREVIOUS ITERATION, BOUNDARIES, AND GIVEN θ, ΔS RECOMPUTE P(ξ, η) AND Q(ξ, η)
ONE SLOR ITERATION STEP FOR x, y
NO
CONVERGED ?
YES
FROM COARSE SOLUTION, INTERPOLATE TO FIND INITIAL CONDITIONS FOR FINE SOLUTION
FINE GRID SOLUTION PROCESS (IDENTICAL TO COARSE SOLUTION PROCESS)
NEW P AND Q
NEW x AND y
STOP

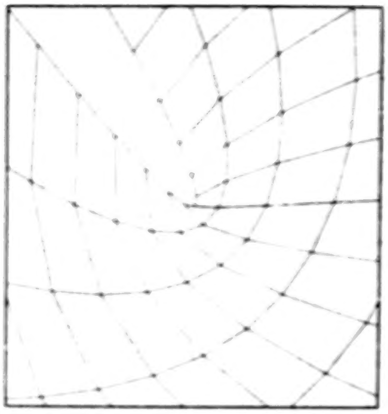
- RESULTS IN SPEEDUP (OVER NORMAL SLOR) BY FACTOR OF 15.

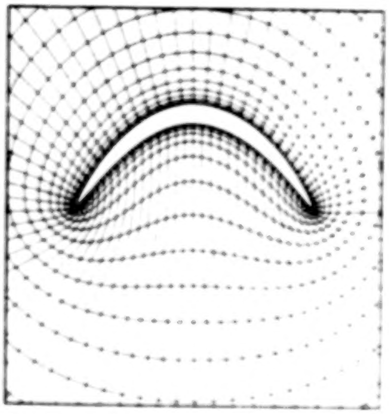
Numerical convergence is greatly accelerated by an additional feature: coarse-fine sequencing. The solution is first iterated to convergence on a coarse grid consisting of every third point in the ξ direction and every third point in the η direction. This convergence requires relatively little computer time since the amount of arithmetic being done per step is one-ninth that which would otherwise be done. The coarse solution is then interpolated to provide initial conditions for a fine solution using all of the points. Coarse-fine sequencing produces a speed-up over normal SLOR by a factor of up to 15. Grids have been generated, for simple cases, in as little as two-thirds of a second of CPU time per thousand grid points on a CDC 7600 computer, including "set-up" overhead.

GRIDS ABOUT HIGHLY CAMBERED ELLIPSE

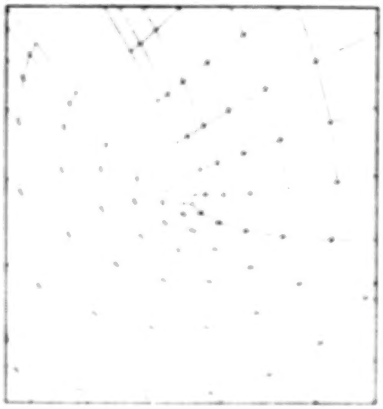


GENERATED BY
LAPLACE
EQUATION,
SHOWING POOR
CONTROL OF CELL
SIZE AND
SKEWNESS



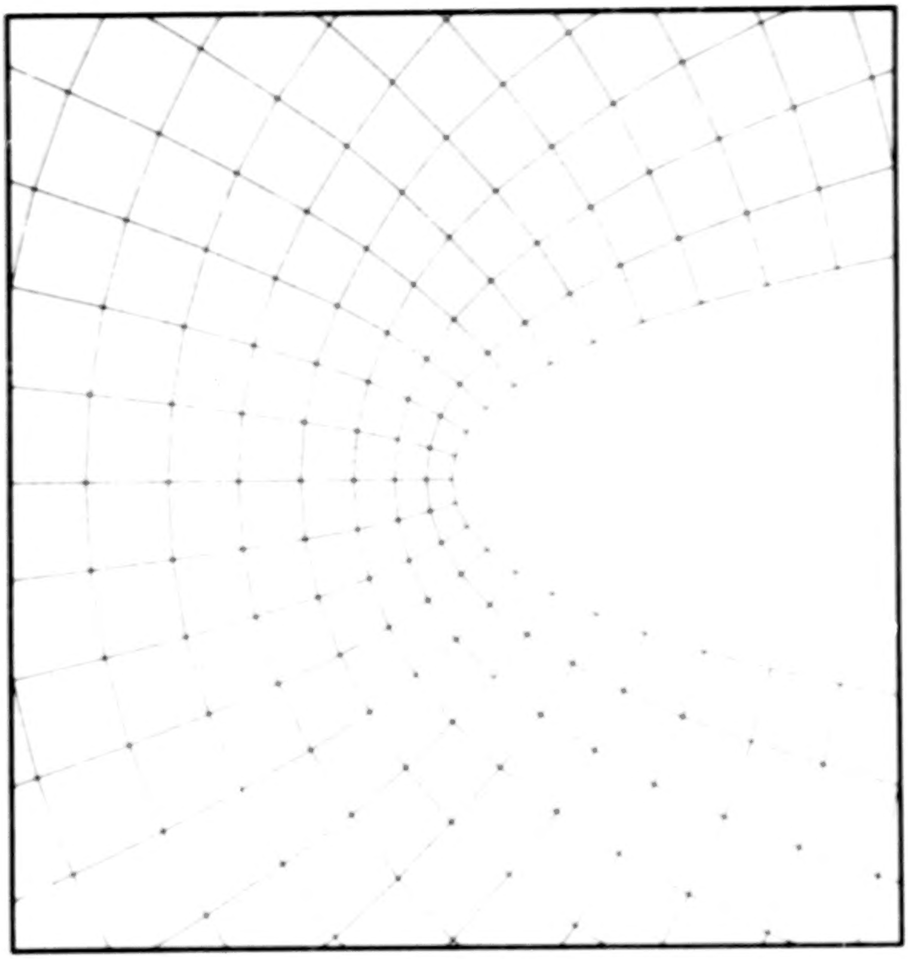


GENERATED BY
POISSON
EQUATION
WITH CONTROL
AT BOUNDARIES



The effectiveness of the grid control is demonstrated in this comparison of two grids about a highly cambered 12:1 elliptical airfoil. In the two top figures is seen a grid generated by the Laplace equations--like the Poisson equations but with P = Q = 0. Uncontrolled cell size and skewness are clearly seen. The figures on the bottom were generated by the present method with the grid control at the boundaries. It was required that the lines intersect the airfoil at 90° and that the standoff distance be 0.005 chord lengths at all points on the airfoil surface. The angle requirement was satisfied to a tolerance of + 0.1° and the distance to a tolerance of ± 0.00001 chord lengths.

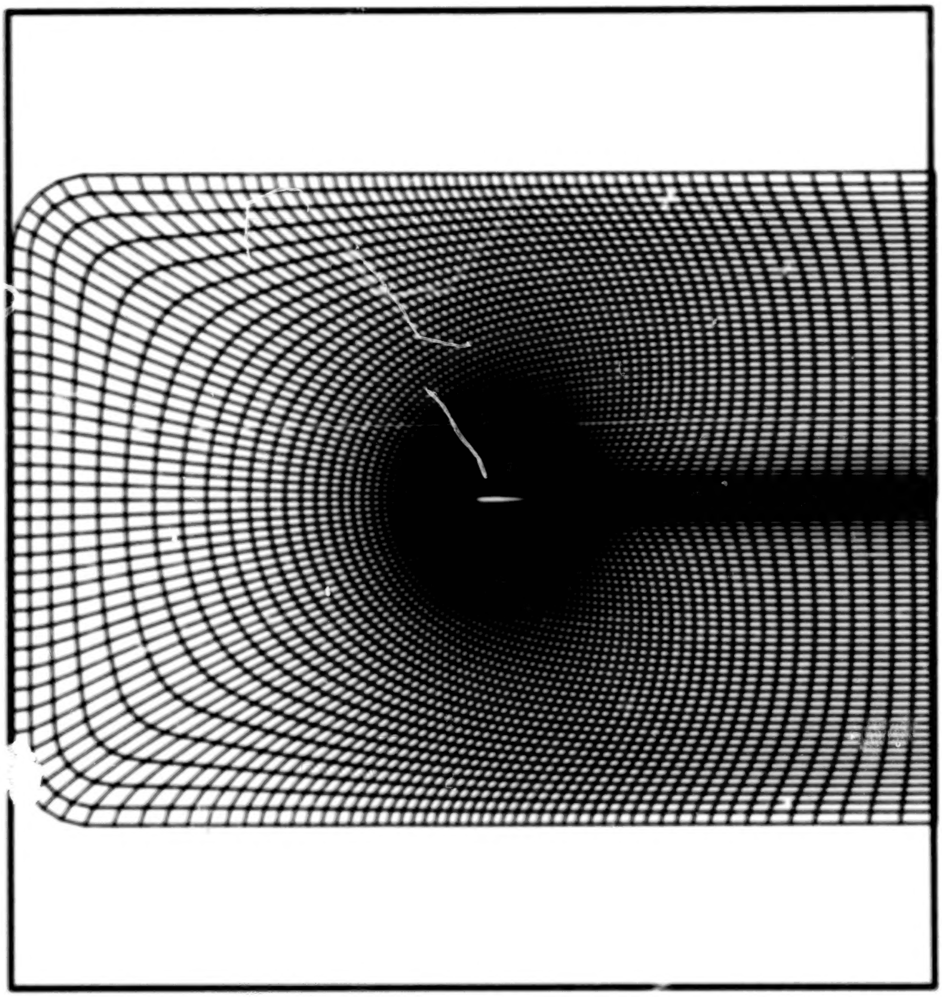
NEARLY SQUARE GRID CELLS ALONG SURFACE IN LEADING EDGE REGION OF NACA 0012 AIRFOIL



GRID SPACING NORMAL TO SURFACE (Δs) SPECIFIED AS EQUAL TO LOCAL VALUE OF ARC-LENGTH ALONG SURFACE

An interesting capability of this method is seen in this close-up view of the leading edge region of a grid about an NACA 0012 airfoil. The angle requirement need not be 90° and it need not be equal at all points on the airfoil surface. Likewise, the standoff spacing need not be constant. In this grid the angle requirement was chosen as 90° everywhere on the airfoil, but the standoff spacing requirement (normal to the airfoil surface) was taken to be equal to the local value of the arc-length along the surface. This produces grid cells along the surface that are nearly square, despite greatly varying surface distribution.

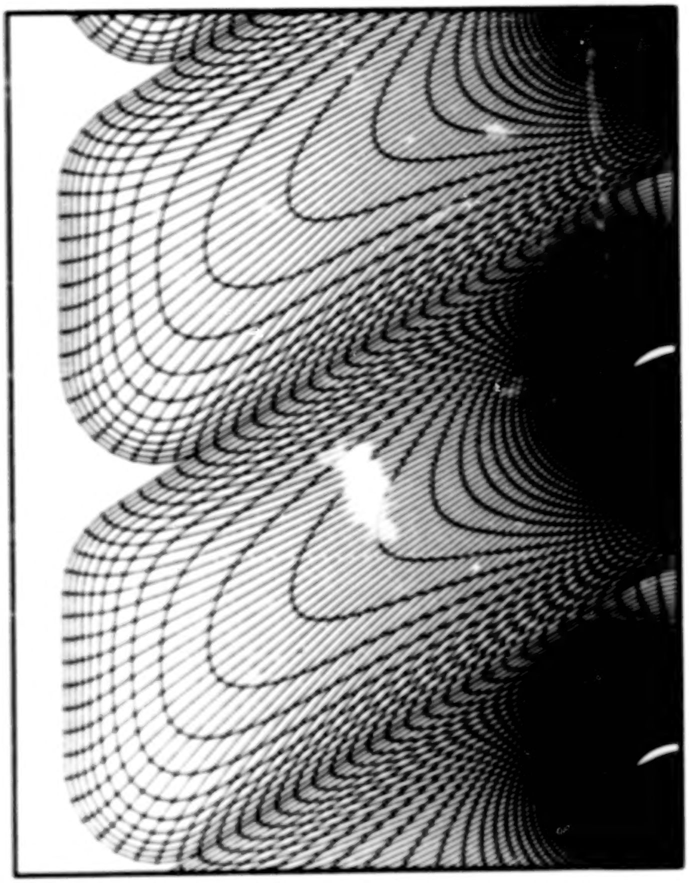
C-TYPE GRID FOR MODELING WIND TUNNEL WALLS



GRID SPACING AND ANGLES CONTROLLED AT
OUTER BOUNDARY

A C-type grid for modeling flow through a wind tunnel is seen here. Grid spacing and angles are controlled at the outer boundary.

UPSTREAM REGION IN O-TYPE CASCADE GRID

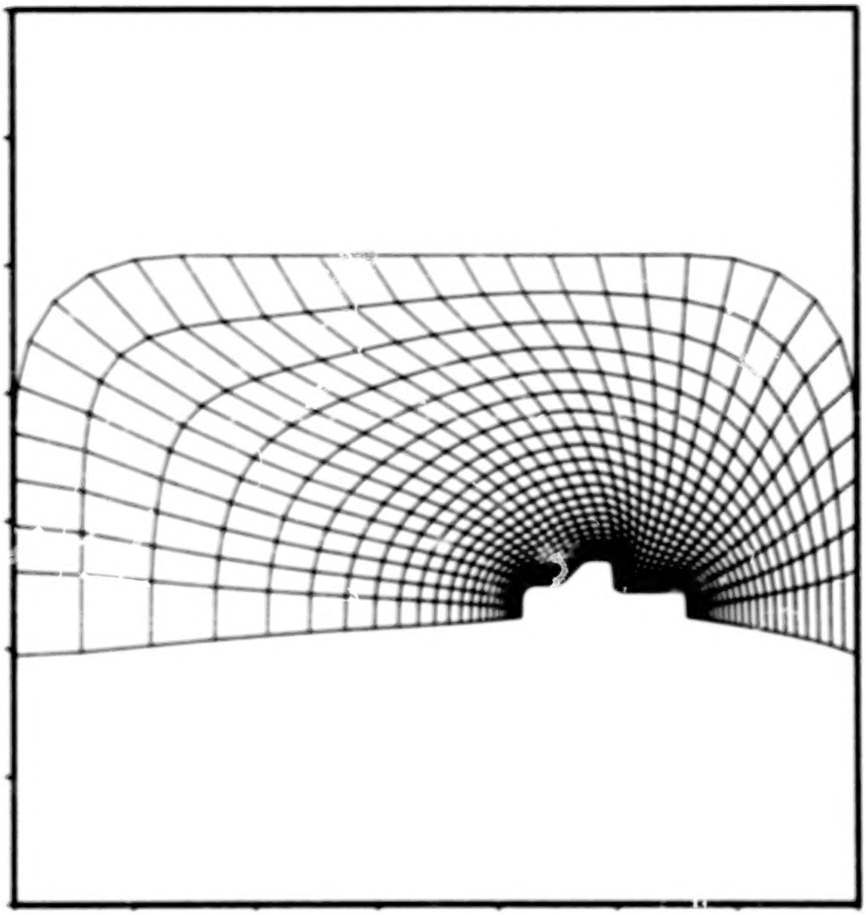


CONTROL OF SPACING AND ANGLES AT OUTER BOUNDARY ENSURES SMOOTH TRANSITION BETWEEN CASCADE ELEMENTS

Control of spacing and angles at outer boundary is applied here to a cascade. It was specified that the lines of constant η which intersect the top and bottom parts of the outer boundary of each cascade element do so vertically. Thus, the application of periodic boundary conditions between cascade elements is facilitated.

This same capability ensures smooth vertical transition across the branch-cut in the wake region of a C-type grid.

GRID USED IN STUDY OF BLAST WAVE ENCOUNTERING STATIONARY PICKUP TRUCK



The versatility of the method is illustrated here.

FEATURES OF GRAPE (GRIDS ABOUT AIRFOILS USING POISSON'S EQUATION), A USER-ORIENTED FORTRAN COMPUTER PROGRAM

- θ, Δs ARE INPUT
- CODING IS MODULAR, GENEROUSLY COMMENTED, SYNTACTICALLY CONSERVATIVE
- BUILT-IN DEFAULT CASE, AND SIMPLE, WELL THOUGHT-OUT INPUT WHICH IS CHECKED BEFORE USE
- VERSATILE: C-TYPE OR O-TYPE, FREE-STREAM OR WIND TUNNEL OR CASCADE, VISCOUS OR INVISCID
- GRAPHICAL OUTPUT
- FAST
- WELL DOCUMENTED AND ACTIVELY SUPPORTED

A user-oriented, well documented, FORTRAN computer program, called GRAPE, has been written to employ this grid generation method. It is available from the Applied Computational Aerodynamics Branch at NASA-Ames Research Center.

BLANK PAGE

BLANK PAGE

USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES

Joseph L. Steger*
Flow Simulations, Inc., Sunnyvale, CA 94036

and

Reese L. Sorenson
NASA Ames Research Center, Moffett Field, CA 94035

Abstract

Interpreting previous work, hyperbolic grid generation procedures are formulated in the style of the elliptic partial differential equation schemes used to form body fitted meshes. For problems in which the outer boundary is not constrained, the hyperbolic scheme can be used to efficiently generate smoothly varying grids with good step size control near the body. Although only two dimensional applications are presented, the basic concepts are shown to extend to three dimensions.

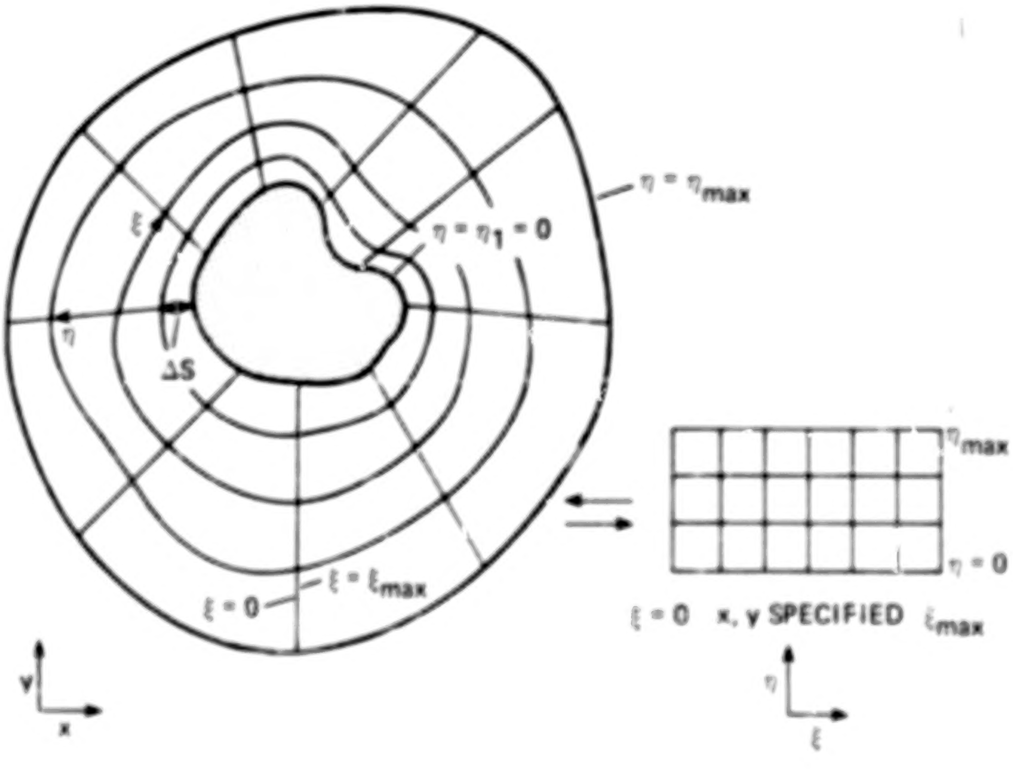
---

*Now with Stanford University.

The task of generating the exterior mesh about an arbitrary closed body as indicated in this slide is undertaken. The location of the outer boundary is not specified; it only need be far removed from the inner boundary. Such a grid generation problem is encountered in external flow aerodynamics.

We seek a grid composed of constant ξ and η lines as indicated in this slide, given initial x,y data along ξ at η = 0. The grid generation equations, just as the flow field equations, are solved in the uniform transform plane.

SKETCH OF PHYSICAL AND COMPUTATIONAL PLANE



$\eta = \eta_{max}$
$\eta = \eta_1 = 0$
ξ
η
ΔS
$\xi = 0$
$\xi = \xi_{max}$
η_{max}
$\eta = 0$
$\xi = 0$ x, y SPECIFIED ξ_{max}
y
x
η
ξ

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B12
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C12
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A13
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B10
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW 377 5/A1
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B5
Youn H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

ATTENDEES 561 7/A3

Partial differential equations are sought to generate a smoothly varying mesh such that grid lines of the same family do not interect or coalesce. Two systems of nonlinear hyperbolic partial differential equations have been considered for the given initial data sketched in the previous slide. As indicated in this slide, these systems each use the condition of orthogonality and a geometric constraint.

HYPERBOLIC GRID GENERATION EQUATIONS

ARC LENGTH-ORTHOGONALITY SCHEME

$$x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2 = (\Delta s)^2$$

$$x_\xi x_\eta + y_\xi y_\eta = 0$$

VOLUME-ORTHOGONALITY SCHEME

$$x_\xi y_\eta - x_\eta y_\xi = V$$

$$x_\xi x_\eta + y_\xi y_\eta = 0$$

IN BOTH CASES $\Delta\xi = \Delta\eta = 1$

The previously described nonlinear partial differential equations must be shown to be properly posed for the given initial value data. As a first step, the equations are cast in a locally linearized form so that they can be analysed as a system of two first order partial differential equations. For the locally linearized form to be meaningful, the equations are expanded about a nearby known solution or state.

LOCALLY LINEARIZED FORM

$$x_\xi x_\eta + y_\xi y_\eta = 0$$

$$x_\xi y_\eta - x_\eta y_\xi = V$$

EXPAND x AND y ABOUT KNOW STATE $\tilde{x}, \tilde{y}$

$$\text{E.G. } x_\xi y_\eta = (\tilde{x} + x-\tilde{x})_\xi\,(\tilde{y} + y-\tilde{y})_\eta$$

$$= \tilde{x}_\xi \tilde{y}_\eta + (x_\xi - \tilde{x}_\xi)\tilde{y}_\eta + (y_\eta - \tilde{y}_\eta)\tilde{x}_\xi + 0(\Delta^2)$$

$$= \tilde{y}_\eta x_\xi + \tilde{x}_\xi y_\eta - \tilde{x}_\xi \tilde{y}_\eta + 0(\Delta^2)$$

OBTAIN LOCALLY LINEARIZED FORM

$$\begin{pmatrix} \tilde{x}_\eta & \tilde{y}_\eta \\ \tilde{y}_\eta & -\tilde{x}_\eta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\xi + \begin{pmatrix} \tilde{x}_\xi & \tilde{y}_\xi \\ -\tilde{y}_\xi & \tilde{x}_\xi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_\eta = \begin{pmatrix} 0 \\ V+\tilde{V} \end{pmatrix}$$

Analysis of the locally linearized partial differential equations indicates that the equations can be marched in η provided that $x_\xi^2 + y_\xi^2 \neq 0$. That is, the grid spacing in ξ cannot be of zero length. The fact that $B^{-1}A$ is a symmetric matrix ensures that it has real, distinct eigenvalues. This then means that the system is hyperbolic if η is used as the marching or time-like direction.

HYPERBOLICITY

LOCALLY LINEARIZED VOLUME-ORTHOGONALITY EQUATION

$$\tilde{A}\vec{r}_\xi + \tilde{B}\vec{r}_\eta = \vec{f}$$

$$\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad A = \begin{pmatrix} x_\eta & y_\eta \\ y_\eta & -x_\eta \end{pmatrix}, \quad B = \begin{pmatrix} x_\xi & y_\xi \\ -y_\xi & x_\xi \end{pmatrix}$$

FIND:

a) B^{-1} EXISTS IF $x_\xi^2 + y_\xi^2 \neq 0$

b) $B^{-1}A$ IS SYMMETRIC

THEREFORE LINEARIZED EQUATIONS ARE HYPERBOLIC

The grid generation equations can be solved using standard numerical techniques for first order systems of hyperbolic partial differential equations. In our case we have used a noniterative implicit finite difference procedure. An unconditionally stable implicit scheme was selected so that an arbitrary mesh step size can be specified in the marching direction. The same kind of numerical procedure is used to solve the flow field equations.

NUMERICAL SOLUTION OF VOLUME-ORTHOGONALITY EQUATIONS

USES IMPLICIT FINITE DIFFERENCE SCHEME FOR

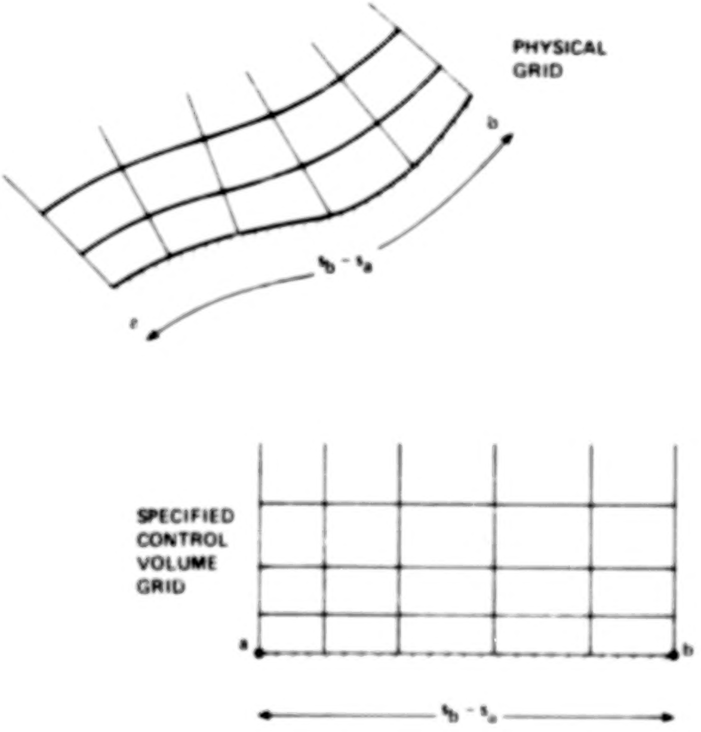
$$x_\xi y_\eta - x_\eta y_\xi = V$$

$$x_\xi x_\eta + y_\xi y_\eta = 0$$

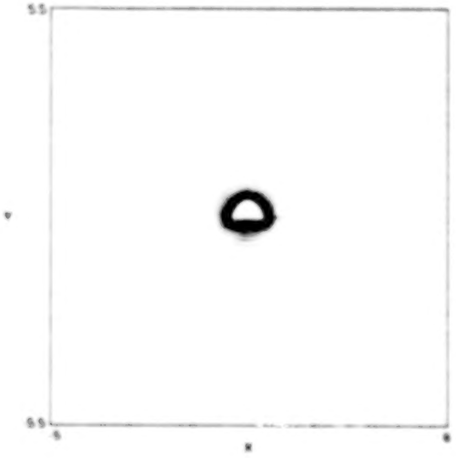
SCHEME IS :
- **a) UNCONDITIONALLY STABLE**
- **b) NONITERATIVE**
- **c) SECOND ORDER IN ξ, FIRST ORDER IN η**
- **d) REQUIRES A BLOCK TRIDIAGONAL INVERSION**

The volume orthogonality scheme requires that the user specify the volume (area in 2-D) of each mesh cell. The quality of the grid will, to a large extent, be determined by the user's cleverness in specifying these volumes. To specify these volumes, we currently define a simple geometric shape (e.g. circle or straight line) which has exactly the same arc length as the body we wish to grid. An algebraically clustered grid is then created by the user for the simple geometric shape. The volumes of this simple grid, the control volume grid, are then used directly on a point by point basis in the hyperbolic grid generation equations.

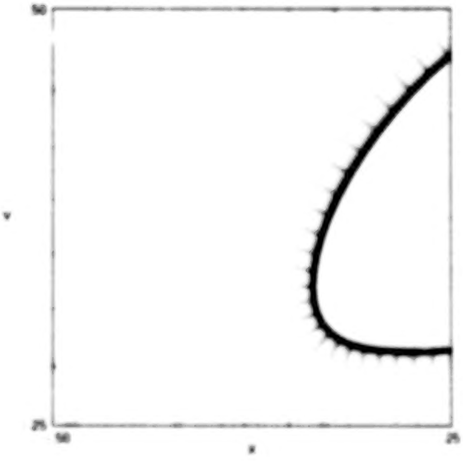
SELECTION OF VOLUMES



These slides show views of a viscous grid generated for a typical cross section of an aircraft fuselage. The control volumes were specified about a circle whose arc length matches that of the fuselage. Initially these control volumes are proportional to the grid-point arc-length spacing around the fuselage. Ultimately, however, control volumes that are uniformly spaced in the circumferential (i.e. θ) direction are specified. Thus in far field a polar coordinate system is formed.

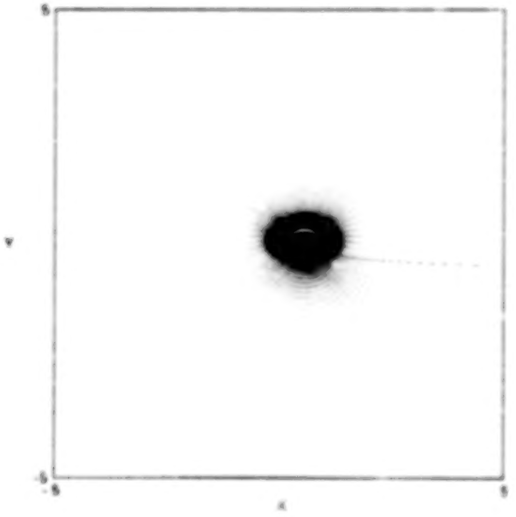


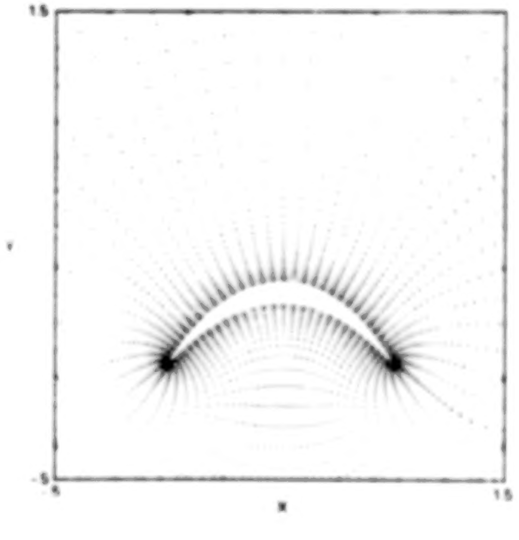
VISCOUS GRID GENERATED ABOUT TYPICAL
AIRCRAFT FUSELAGE CROSS SECTION
OVERVIEW



GRID DETAIL NEAR CORNER

This slide shows views of an inviscid grid generated about a highly cambered profile. The same type of control volume used previously is employed. A uniform grid spacing was specified in the direction away from the body as is clear from the view showing grid detail near the body.

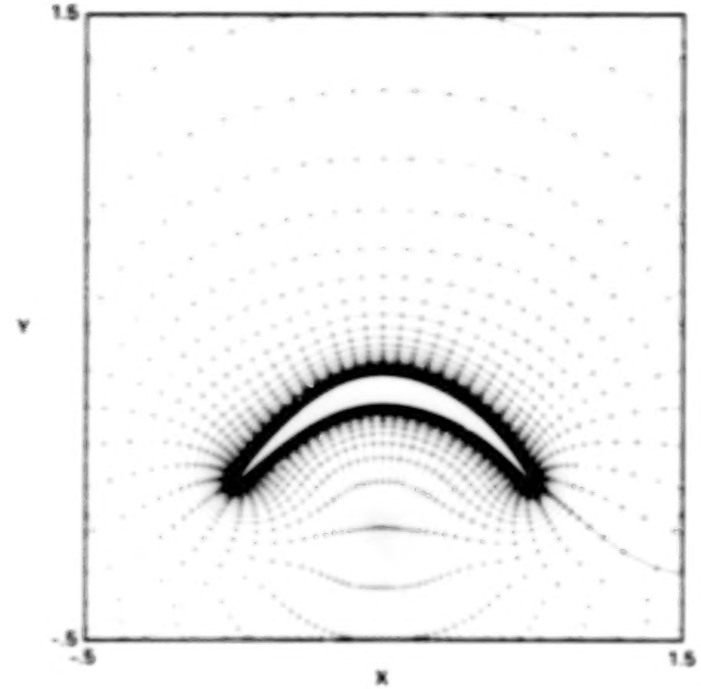

INVISCID GRID GENERATED ABOUT HIGHLY CAMBERED AIRFOIL
OVERVIEW


GRID DETAIL OF BLADE

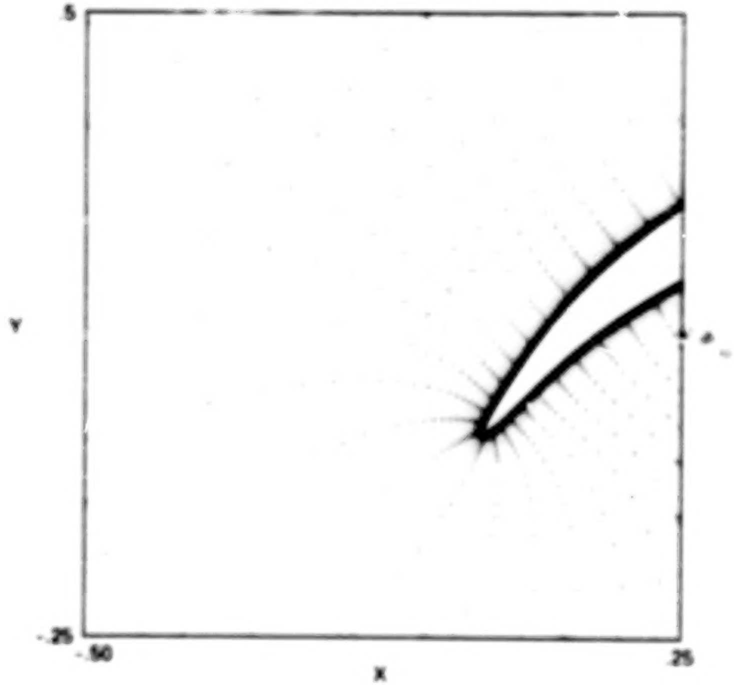
In this case a viscous grid is generated about the cambered profile. The normal grid spacing at the body is 0.01% of the chord length. Note that because volume is specified the grid spacing grows in the marching direction so as to prevent the circumferential spacing from vanishing. For a profile with twice the camber, however, this process breaks down and grid lines do coelesce. In these cases a more sophiticated means of specifying the volumes is needed.

VISCOUS GRID GENERATED ABOUT HIGHLY CAMBERED AIRFOIL

GRID DETAIL NEAR BODY

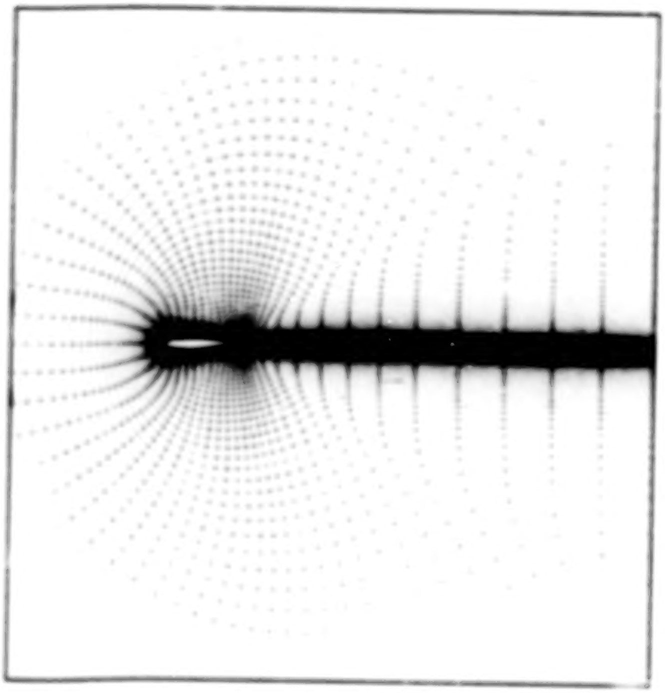


GRID DETAIL NEAR LEADING EDGE

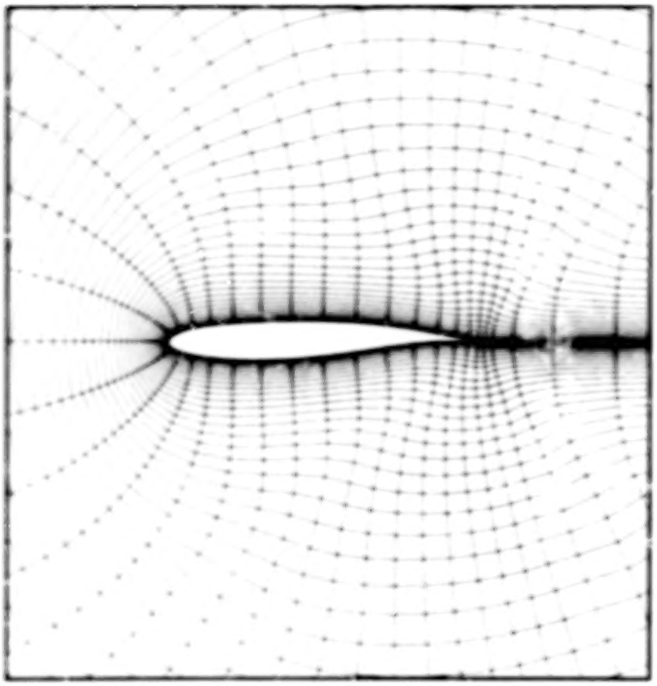


These views show the hyperbolic grid generation scheme applied to generating a "C-grid" about a cambered airfoil. Here the control volume grid is generated about a straight line, that is, it is nothing more than a clustered rectangular grid. It is clear from the view at the trailing edge that some adjustments are needed to the current numerical treatment of discontinuous boundary data.

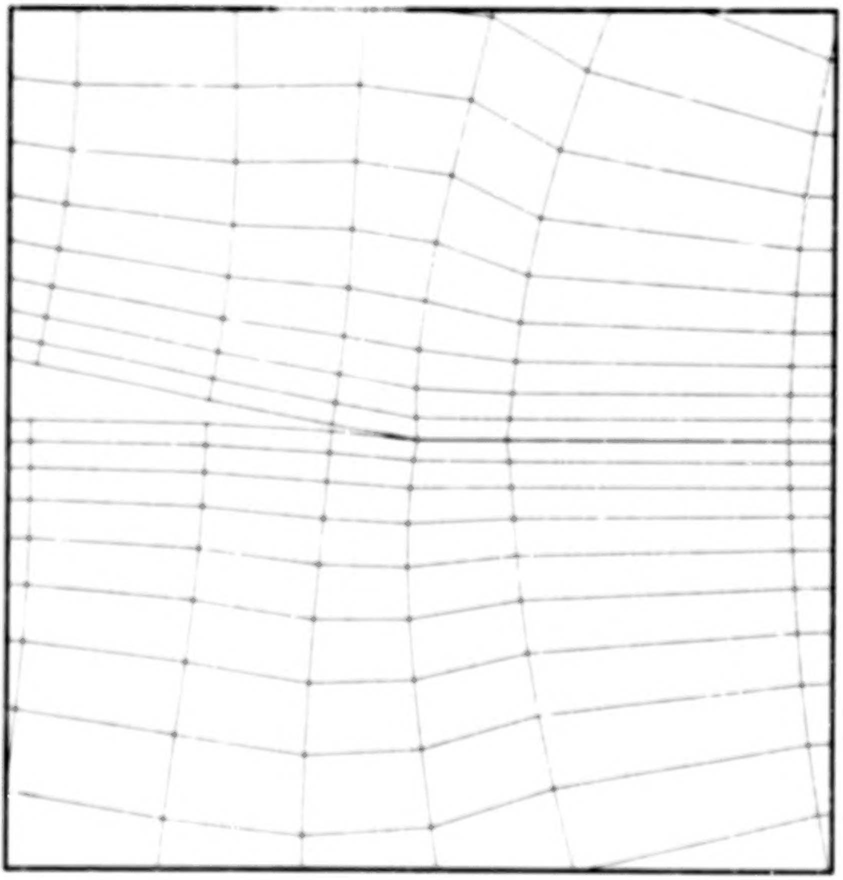
GENERATION OF C-GRID ABOUT
CAMBERED AIRFOIL
OVERVIEW



GRID DETAIL NEAR BODY



GRID DETAIL AT TRAILING EDGE



HYPERBOLIC GRID GENERATION ADVANTAGES

- SMOOTHLY VARYING GRID IS FOUND
- GOOD USER CONTROL OF CLUSTERING NEAR BOUNDARY
- FAST GRID GENERATION
- ORTHOGONAL OR NEARLY ORTHOGONAL
- AUTOMATICALLY TREATS COMPLEX SHAPES

HYPERBOLIC GRID GENERATION DISADVANTAGES

- OUTER BOUNDARY CANNOT BE SPECIFIED (UNLESS ITERATIVE SHOOTING METHOD DEVISED)
- SCHEME TENDS TO PROPAGATE DISCONTINUOUS BOUNDARY DATA
- POORLY SPECIFIED BOUNDARY DATA AND CONTROL VOLUMES CAN RESULT IN "SHOCK-WAVE" LIKE BREAKDOWN

The hyperbolic grid generation scheme can also be formulated in three dimensions. With volume specified as one constraint, orthogonality can only be enforced in two of the coordinate directions. The three partial differential equations shown form a hyperbolic system for marching in ζ. Proof that the equations are hyperbolic was quite tedious, required considerable insight, and was carried out by Dennis Jespersen of Oregon State University.

EXTENSION TO THREE DIMENSIONS – VOLUME AND TWO ORTHOGONALITY

$$\frac{d\vec{r}}{d\xi} \cdot \frac{d\vec{r}}{d\zeta} = 0$$

$$\frac{d\vec{r}}{d\eta} \cdot \frac{d\vec{r}}{d\zeta} = 0$$

$$\left| \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} \right| = V$$

SYSTEM IS FOUND TO BE HYPERBOLIC

The three constraints of orthogonality do not form a hyperbolic system of partial differential equations. Neither are the equations of elliptic type. In fact, their classification and what if any type of boundary data makes them unique is unknown to the authors.

EXTENSION TO THREE DIMENSIONS –

THREE ORTHOGONALITY

$$\frac{d\vec{r}}{d\xi} \cdot \frac{d\vec{r}}{d\zeta} = 0$$

$$\frac{d\vec{r}}{d\xi} \cdot \frac{d\vec{r}}{d\eta} = 0$$

$$\frac{d\vec{r}}{d\eta} \cdot \frac{d\vec{r}}{d\zeta} = 0$$

SYSTEM CANNOT BE MARCHED AND IS NOT ELLIPTIC

CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS

Frank B. Tatom, Engineering Analysis, Inc.
William R. Waldrop, Tennessee Valley Authority
S. Ray Smith, Engineering Analysis, Inc.

CENTERLINE INTRODUCTION

In order to effectively analyze the flow in sinuous river channels a curvilinear grid system must be developed for use in the appropriate hydrodynamic code. The CENTERLINE program has been designed to generate a two-dimensional grid for this purpose.

The Cartesian coordinates of a series of points along the boundaries of the sinuous channel represent the primary input to CENTERLINE. The program calculates the location of the river centerline, the distance downstream along the centerline, and both radius of curvature and channel width, as a function of such distance downstream. These parameters form the basis for the generation of the curvilinear grid.

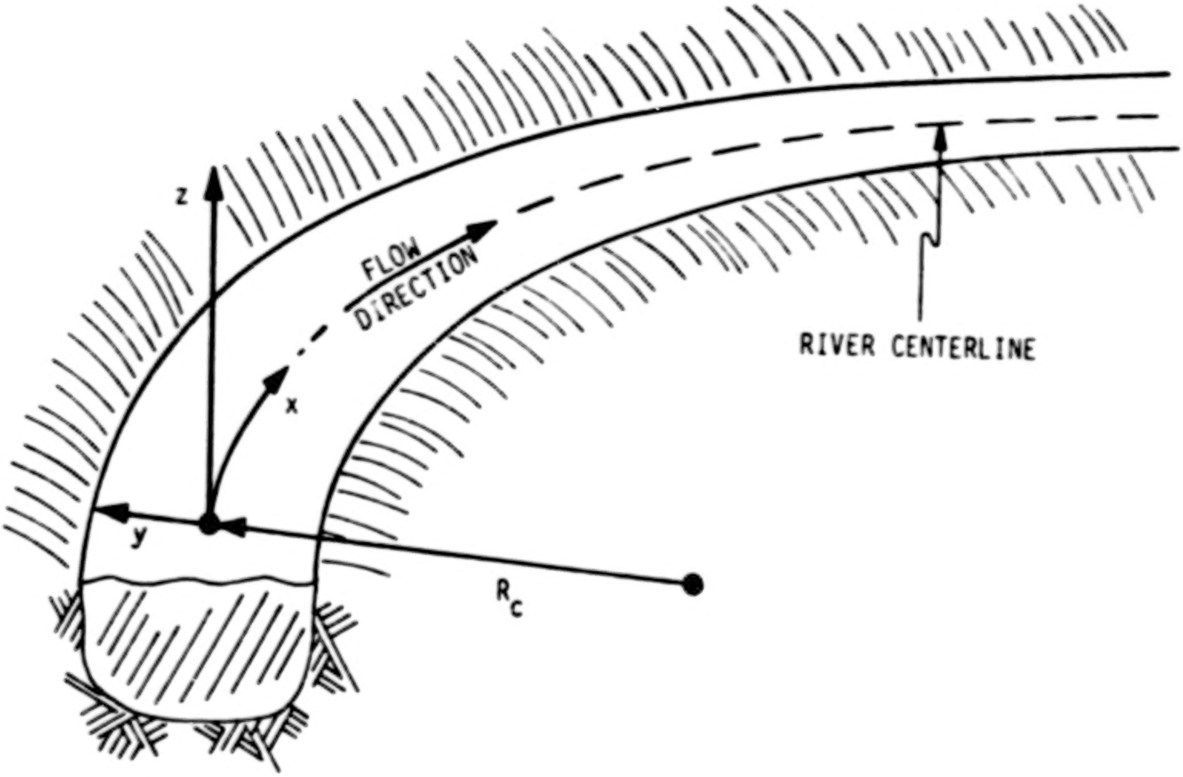
Based on input values for longitudinal and lateral grid spacing, the corresponding grid system is generated and a file is created containing the appropriate parameters for use in the associated explicit finite difference hydrodynamic programs. Because of the option for a nonuniform grid, grid spacing can be concentrated in areas containing the largest flow gradients.

For the case of sinuous channels of constant or nearly constant width the resulting curvilinear grid is orthogonal. The grid generation procedure also provides for dividing the overall flow area under consideration into a series of regions connected along common boundaries. This concept of multiple regions tends to improve computational efficiency.

For many sinuous channels the assumption of constant width is not appropriate. In such situations CENTERLINE generates a nonorthogonal grid which takes into account the nonuniform channel width.

The CENTERLINE program is currently operational and has been used successfully in conjunction with both two- and three-dimensional incompressible hydrodynamic programs. To the authors' knowledge, it is the only *curvilinear* grid program currently coupled with *operational* incompressible hydrodynamic programs for computing two- and three-dimensional river flows.

BASIC CURVILINEAR COORDINATE SYSTEM


z
FLOW DIRECTION
RIVER CENTERLINE
x
y
R_c

GOVERNING EQUATIONS FOR INCOMPRESSIBLE CURVILINEAR FLOW

CONTINUITY:

$$\frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z u) + \frac{\partial}{\partial y}(h_z h_x v) + \frac{\partial}{\partial z}(h_x h_y w)\right] = 0$$

X-MOMENTUM:

$$\rho\left[\frac{\partial u}{\partial t} + \frac{u}{h_x}\frac{\partial u}{\partial x} + \frac{v}{h_y}\frac{\partial u}{\partial y} + \frac{w}{h_z}\frac{\partial u}{\partial z}\right.$$

$$\left. - v\left(\frac{v}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{u}{h_x h_y}\frac{\partial h_x}{\partial y}\right) + w\left(\frac{u}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{w}{h_z h_x}\frac{\partial h_z}{\partial x}\right)\right]$$

$$= \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xx}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yx}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zx})\right]$$

$$+ \frac{\sigma_{xy}}{h_x h_y}\frac{\partial h_x}{\partial y} + \frac{\sigma_{zx}}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{\sigma_{yy}}{h_x h_y}\frac{\partial h_y}{\partial x} - \frac{\sigma_{zz}}{h_x h_z}\frac{\partial h_z}{\partial x} + F_x$$

Y-MOMENTUM:

$$\rho\left[\frac{\partial v}{\partial t} + \frac{u}{h_x}\frac{\partial v}{\partial x} + \frac{v}{h_y}\frac{\partial v}{\partial y} + \frac{w}{h_z}\frac{\partial v}{\partial z}\right.$$

$$\left. - w\left(\frac{w}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{v}{h_y h_z}\frac{\partial h_y}{\partial z}\right) + u\left(\frac{v}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{u}{h_x h_y}\frac{\partial h_x}{\partial y}\right)\right]$$

$$= \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xy}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yy}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zy})\right]$$

$$+ \frac{\sigma_{yz}}{h_y h_z}\frac{\partial h_y}{\partial z} + \frac{\sigma_{xy}}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{\sigma_{zz}}{h_y h_z}\frac{\partial h_z}{\partial y} - \frac{\sigma_{xx}}{h_y h_x}\frac{\partial h_x}{\partial y} + F_y$$

Z-MOMENTUM:

$$\rho\left[\frac{\partial w}{\partial t} + \frac{u}{h_x}\frac{\partial w}{\partial x} + \frac{v}{h_y}\frac{\partial w}{\partial y} + \frac{w}{h_z}\frac{\partial w}{\partial z}\right.$$

$$\left. - u\left(\frac{u}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{w}{h_z h_x}\frac{\partial h_z}{\partial x}\right) + v\left(\frac{w}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{v}{h_y h_z}\frac{\partial h_y}{\partial z}\right)\right]$$

$$= \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xz}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yz}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zz})\right]$$

$$+ \frac{\sigma_{zx}}{h_x h_z}\frac{\partial h_z}{\partial x} + \frac{\sigma_{yz}}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{\sigma_{xx}}{h_z h_x}\frac{\partial h_x}{\partial z} - \frac{\sigma_{yy}}{h_z h_y}\frac{\partial h_y}{\partial z} + F_z$$

ENERGY:

$$\frac{\partial T}{\partial t} + \frac{u}{h_x}\frac{\partial T}{\partial x} + \frac{v}{h_y}\frac{\partial T}{\partial y} + \frac{w}{h_z}\frac{\partial T}{\partial z}$$

$$= \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}\left(\frac{\alpha_x h_y h_z}{h_x}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\alpha_y h_z h_x}{h_y}\frac{\partial T}{\partial y}\right)\right.$$

$$\left. + \frac{\partial}{\partial z}\left(\frac{\alpha_z h_x h_y}{h_z}\frac{\partial T}{\partial z}\right)\right]$$

COMPUTATION OF METRIC COEFFICIENTS

FUNDAMENTAL CONSIDERATIONS:

- APPEAR IN GOVERNING EQUATIONS
- ONLY h_x REQUIRES COMPUTATION
- EVALUATED FOR EACH GRID POINT
- DERIVATIVES ALSO REQUIRED

BASIC RELATIONS:

$$h_x = \frac{R_c + y}{R_c}$$

$$h_y = 1$$

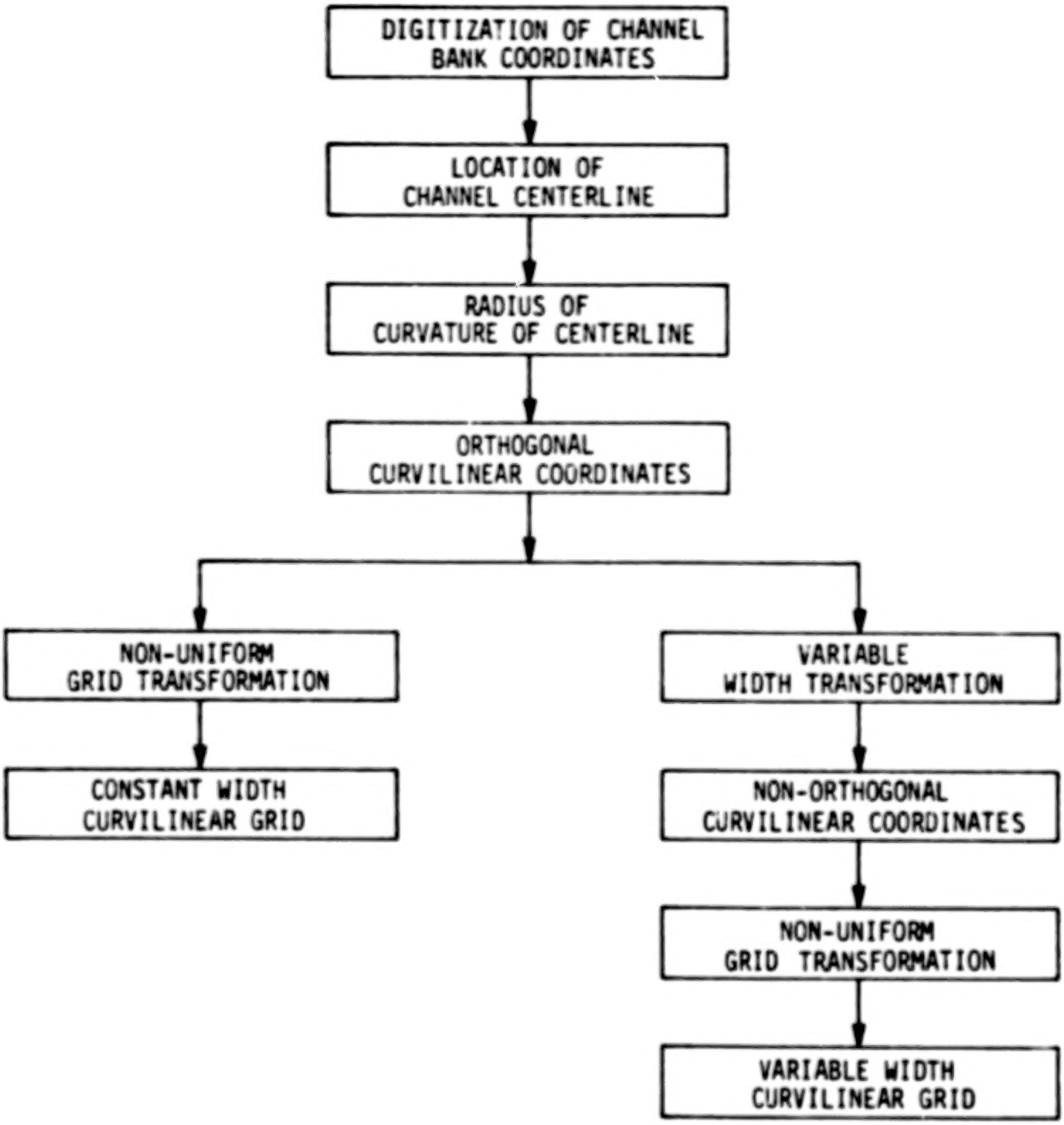
$$h_z = 1$$

DERIVATIVES:

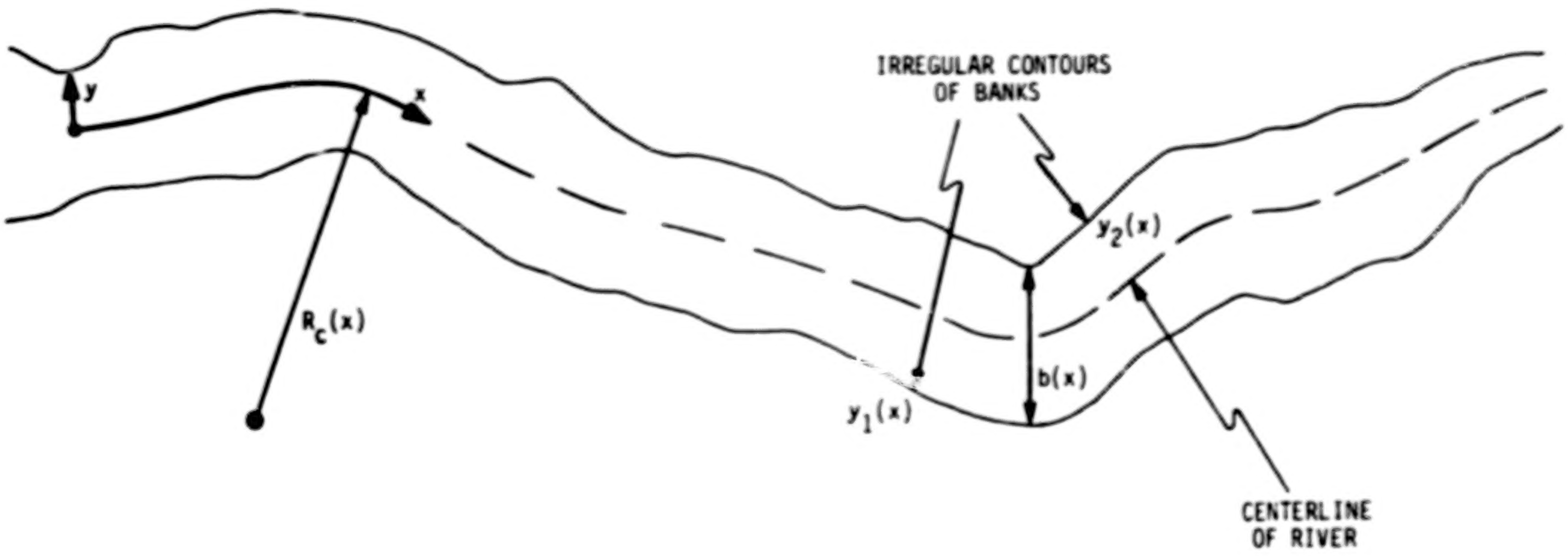
$$\frac{\partial h_x}{\partial x} = - \frac{y}{R_c^2} \frac{dR_c}{dx}$$

$$\frac{\partial h_x}{\partial y} = \frac{1}{R_c}$$

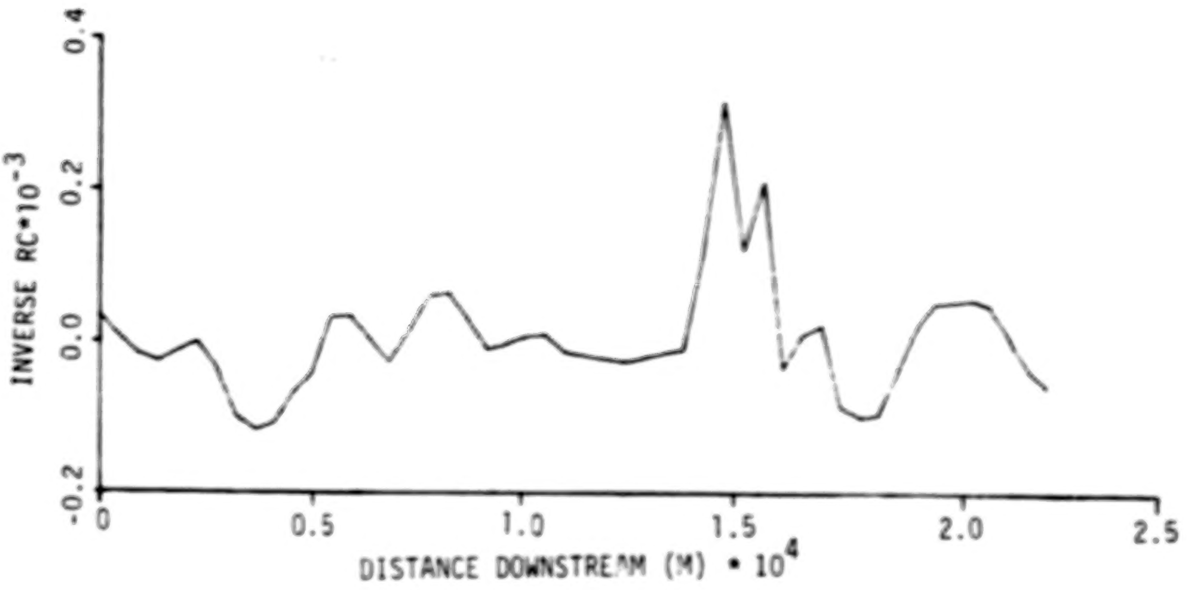
GENERATION OF CURVILINEAR GRIDS


DIGITIZATION OF CHANNEL BANK COORDINATES
LOCATION OF CHANNEL CENTERLINE
RADIUS OF CURVATURE OF CENTERLINE
ORTHOGONAL CURVILINEAR COORDINATES
NON-UNIFORM GRID TRANSFORMATION
CONSTANT WIDTH CURVILINEAR GRID
VARIABLE WIDTH TRANSFORMATION
NON-ORTHOGONAL CURVILINEAR COORDINATES
NON-UNIFORM GRID TRANSFORMATION
VARIABLE WIDTH CURVILINEAR GRID

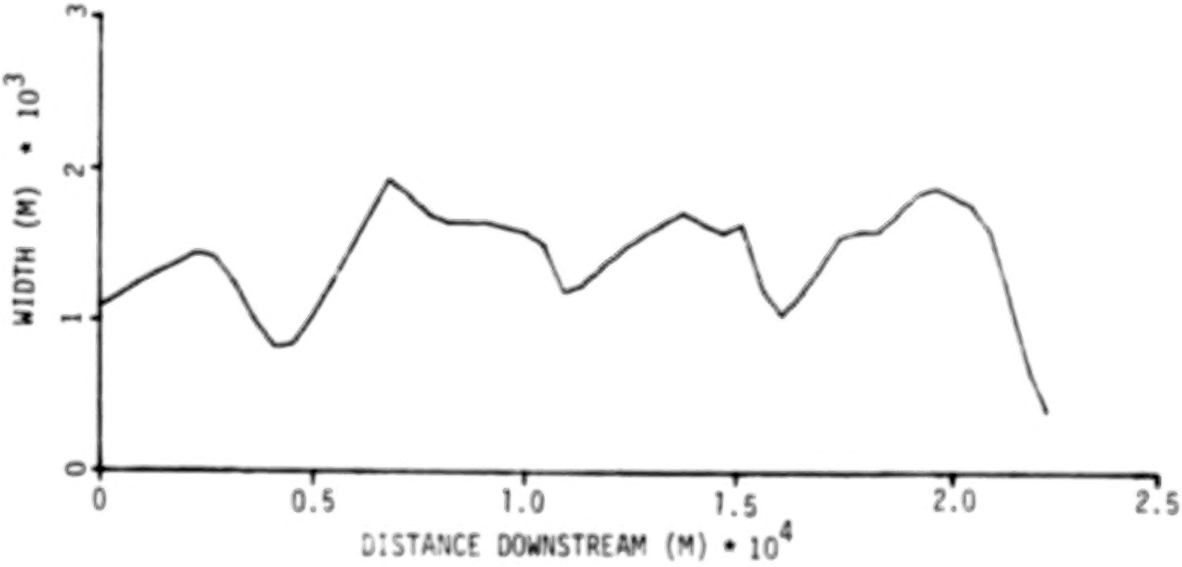
TYPICAL RESERVOIR


y
x
IRREGULAR CONTOURS
OF BANKS
$y_2(x)$
$R_c(x)$
$b(x)$
$y_1(x)$
CENTERLINE
OF RIVER

INVERSE RADIUS OF CURVATURE VS DISTANCE DOWNSTREAM FOR TYPICAL RESERVOIR



CHANNEL WIDTH VS DISTANCE DOWNSTREAM FOR TYPICAL RESERVOIR



COMPUTATION OF RADIUS OF CURVATURE AND CHANNEL WIDTH

- DIGITIZE CARTESIAN COORDINATES OF CHANNEL BANKS
- LOCATE GEOMETRIC CENTERLINE
- COMPUTE DISTANCE ALONG CENTERLINE, x
- COMPUTE RADIUS OF CURVATURE, $R_c(x)$
- COMPUTE CHANNEL WIDTH, $b(x)$

VARIABLE WIDTH TRANSFORMATION

BASIC TRANSFORMATION:

$$b = y_2(x) - y_1(x)$$

$$X = x$$

$$Y = y/b$$

TRANSFORMATION DERIVATIVES:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial X} + \frac{\partial f}{\partial Y} Y'$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial Y} \frac{1}{b}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial X^2} + 2 \frac{\partial^2 f}{\partial X \partial Y} Y' + \frac{\partial^2 f}{\partial Y^2} (Y')^2 + \frac{\partial f}{\partial Y} Y''$$

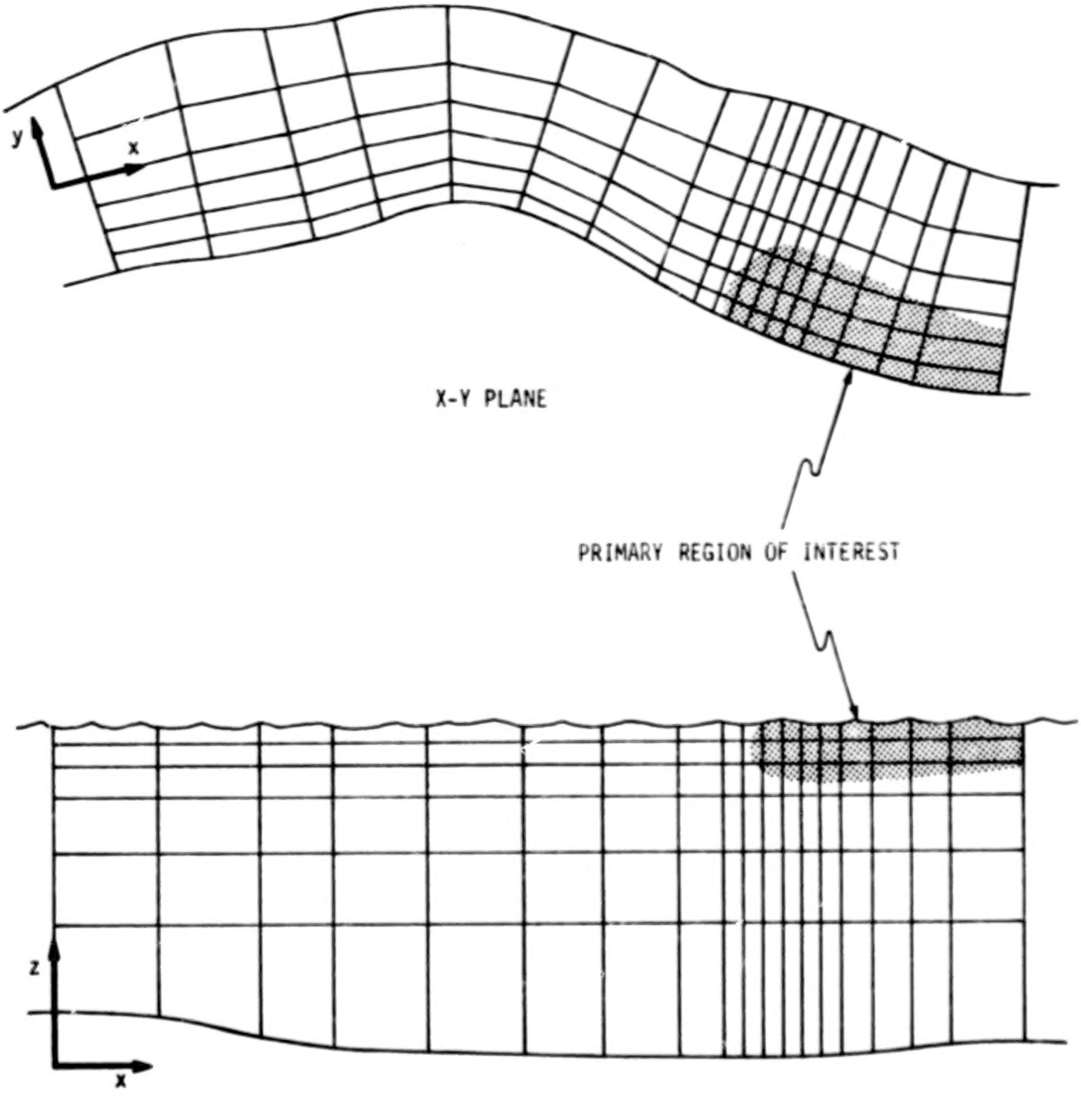
$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial Y^2} \frac{1}{b^2}$$

where

$$Y' = -\frac{Y}{b} \frac{db}{dx}$$

$$Y'' = \frac{2Y}{b^2} \frac{db}{dx} - \frac{Y}{b} \frac{d^2 b}{dx^2}$$

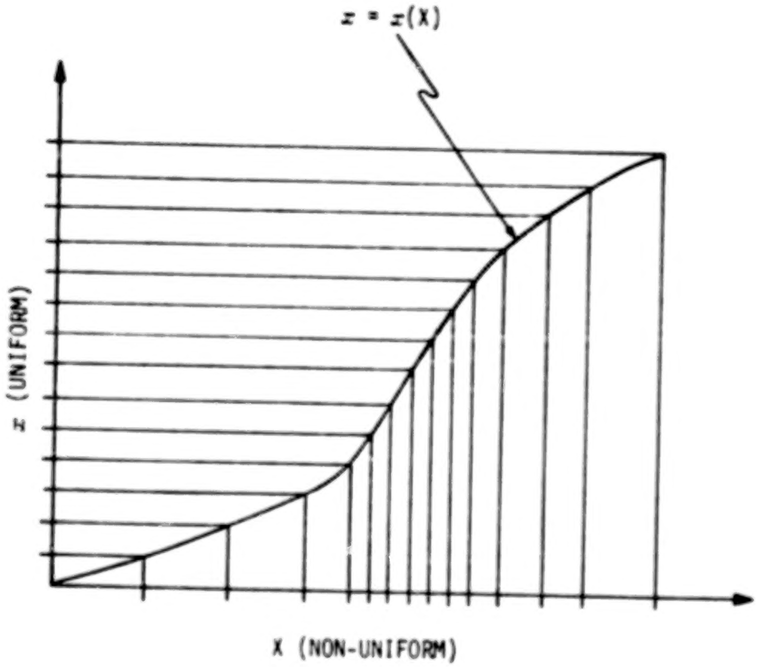
NON-UNIFORM GRID SYSTEM


y
x
PRIMARY REGION OF INTEREST
z
x

X-Y PLANE

X-Z PLANE

RELATIONSHIP BETWEEN NON-UNIFORM AND UNIFORM GRIDS


$z = z(X)$
z (UNIFORM)
X (NON-UNIFORM)

TRANSFORMATION FROM NON-UNIFORM TO UNIFORM GRID

PROCEDURE:

- IDENTIFY "REGIONS OF INTEREST"
- INPUT DESIRED GRID SPACING
- GENERATE TRANSFORMATION DERIVATIVES

BASIC TRANSFORMATION:

$$\left.\begin{array}{l} x = x(X) \\ y = y(Y) \\ z = z(Z) \end{array}\right\} \text{ANALYTICAL TRANSFORMATION FUNCTIONS NOT REQUIRED}$$

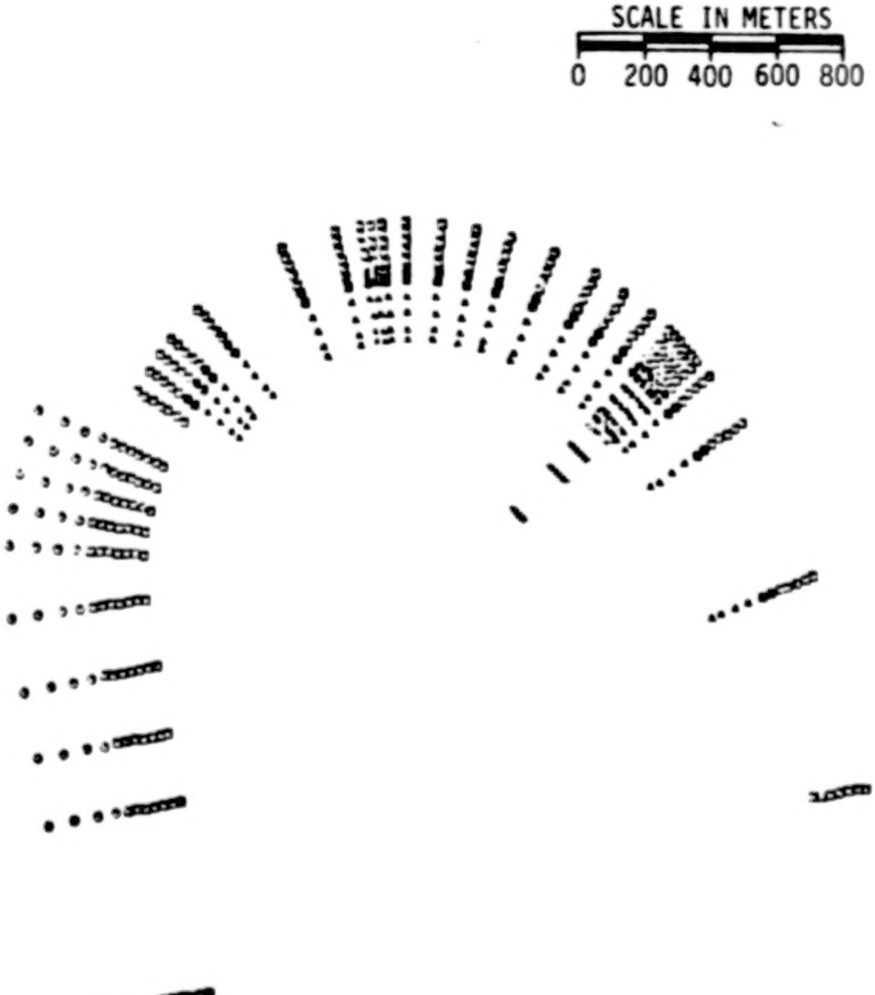
TRANSFORMATION DERIVATIVES:

$$\frac{\partial g}{\partial X} = \frac{\partial g}{\partial x}\frac{\partial x}{\partial X}$$

$$\frac{\partial^2 g}{\partial X^2} = \frac{\partial g}{\partial x}\frac{\partial^2 x}{\partial X^2} + \frac{\partial^2 g}{\partial x^2}\left(\frac{\partial x}{\partial X}\right)^2$$

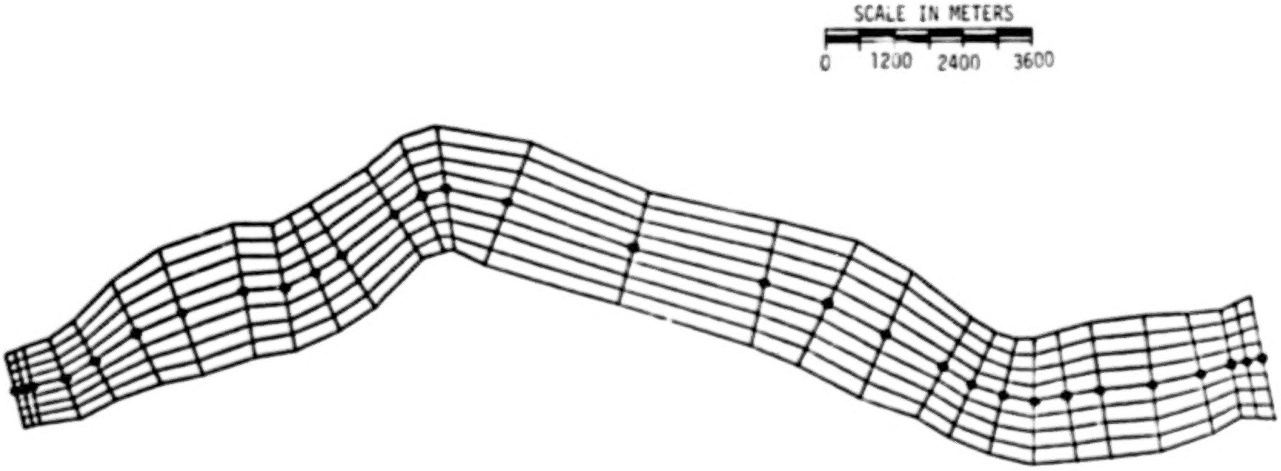
CURVILINEAR GRID FOR CUMBERLAND RIVER SEGMENT

- NEAR TVA GALATIN STEAM PLANT
- CONSTANT WIDTH CHANNEL
- NON-UNIFORM GRID (x, y, & z)
- 4 CONNECTED REGIONS
- USED IN 3-D FLOW COMPUTATIONS


SCALE IN METERS
0
200
400
600
800

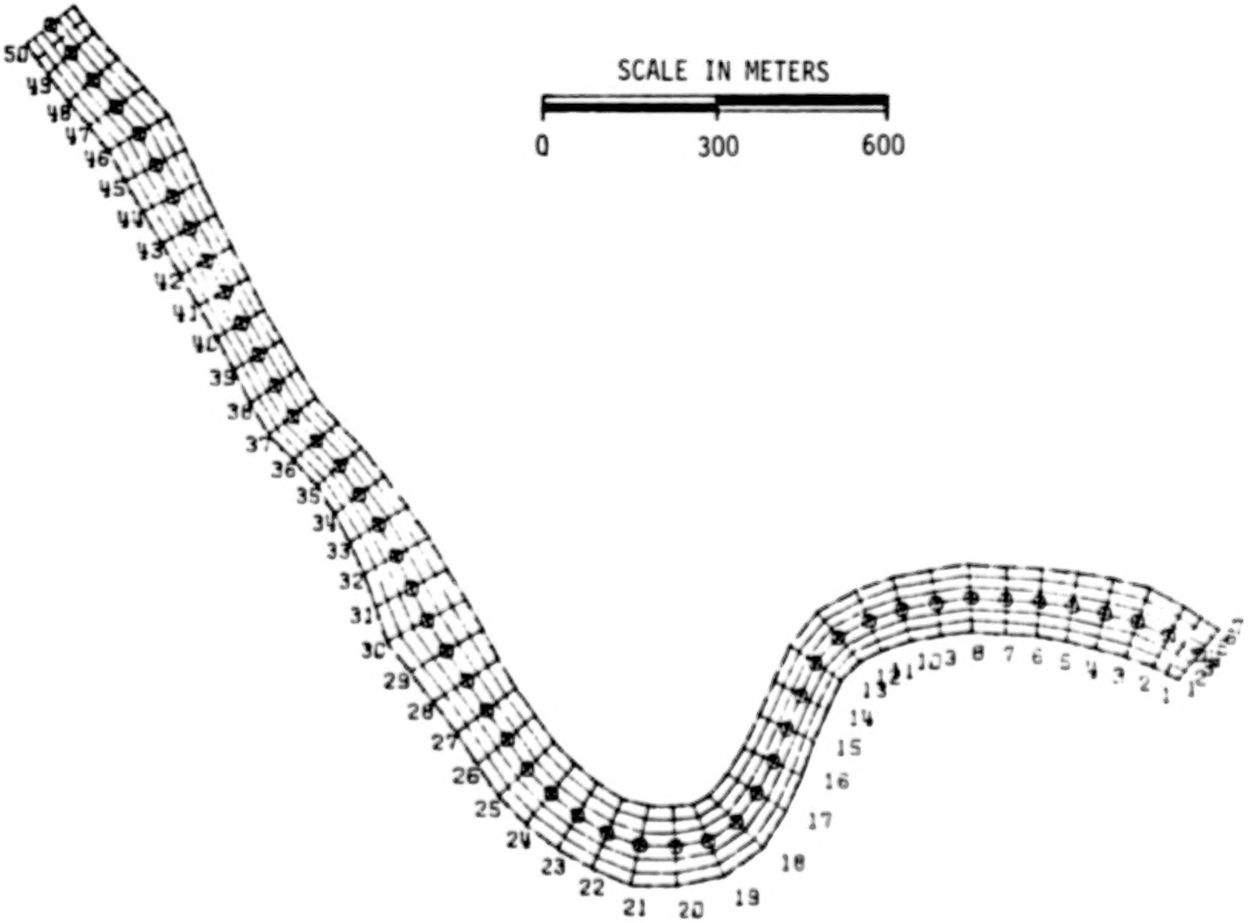
CURVILINEAR GRID FOR TENNESSEE RIVER, WILSON RESERVOIR

- BETWEEN WHEELER AND WILSON DAMS
- VARIABLE WIDTH CHANNEL
- NON-UNIFORM GRID (x only)
- USED IN 2-D DEPTH-AVERAGED FLOW COMPUTATION


SCALE IN METERS
0
1200
2400
3600

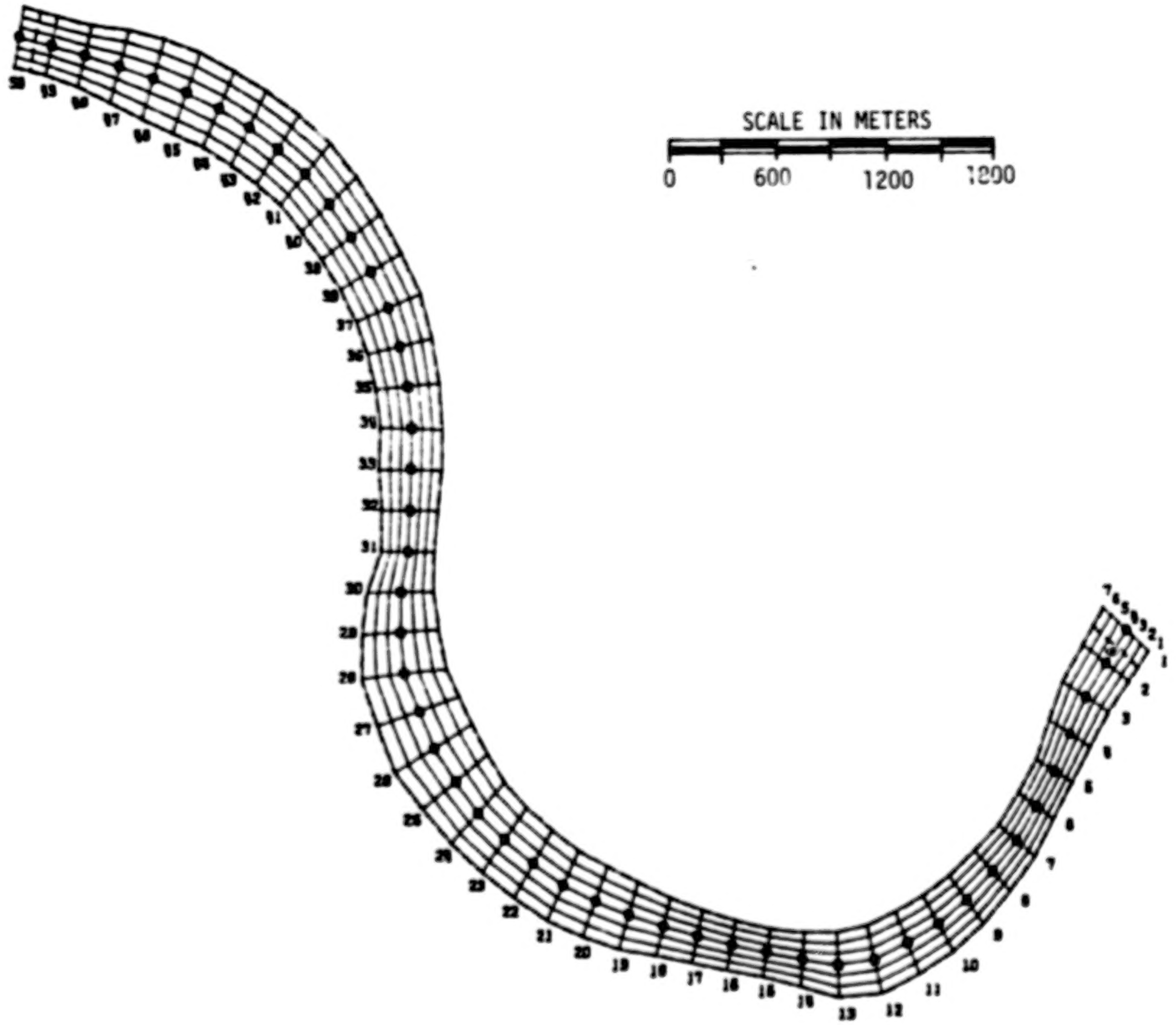
CURVILINEAR GRID FOR GREEN RIVER SEGMENT

- NEAR PARADISE STEAM PLANT
- MODERATE SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID


SCALE IN METERS
0
300
600
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

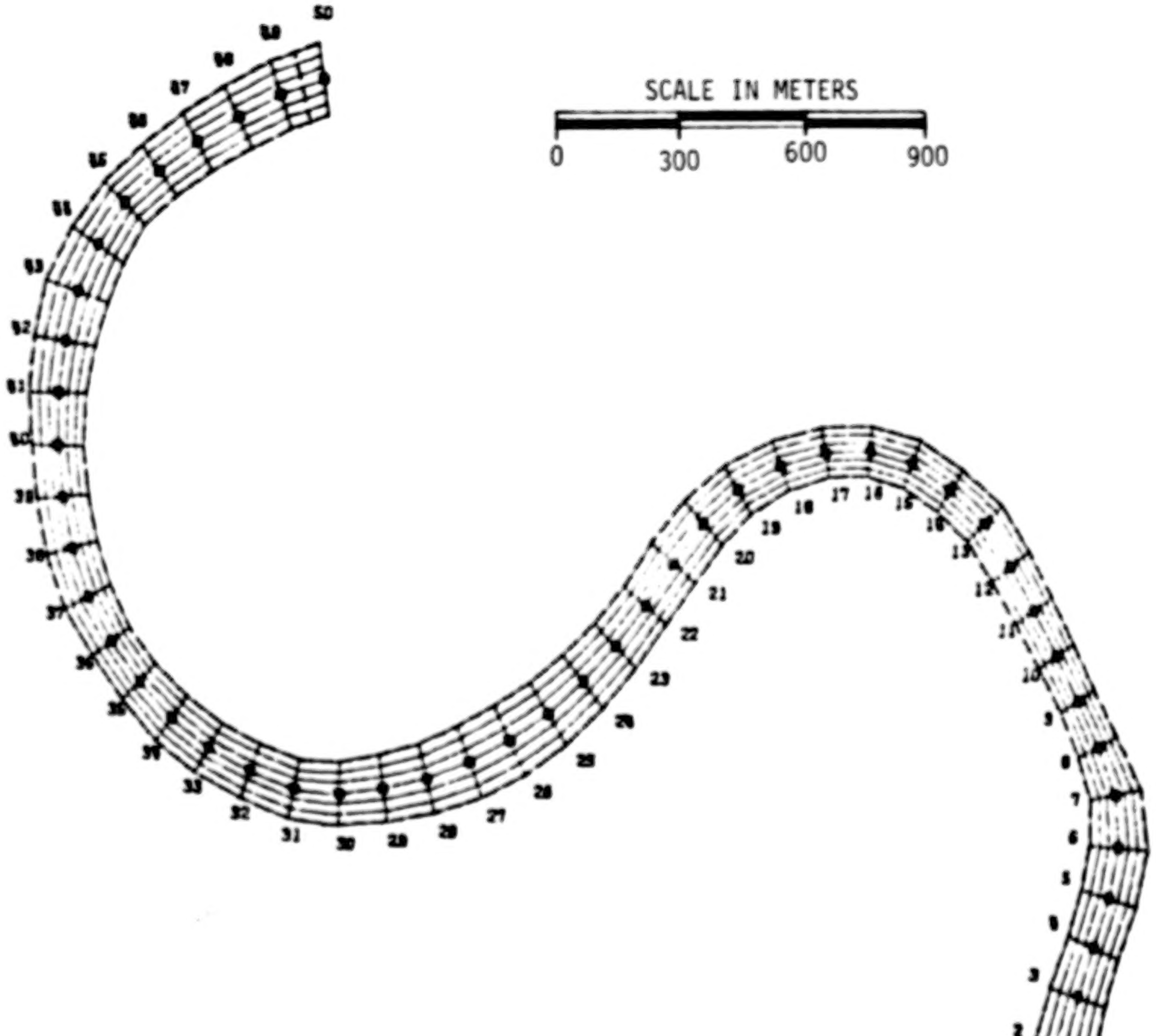
CURVILINEAR GRID FOR TENNESSEE RIVER, WHEELER RESERVOIR

- NEAR REDSTONE ARSENAL
- MODERATE SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID


SCALE IN METERS
0
600
1200
1800

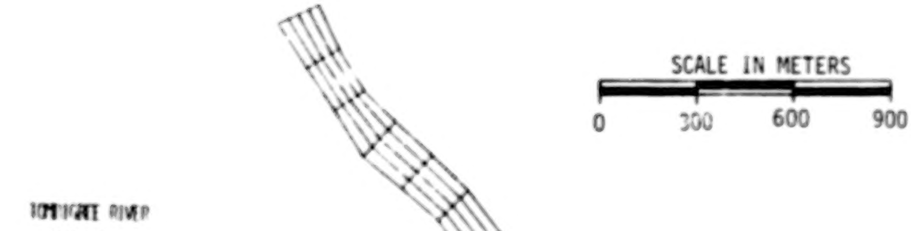
CURVILINEAR GRID FOR LITTLE TENNESSEE RIVER SEGMENT

- PART OF TELLICO LAKE
- HIGH SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID

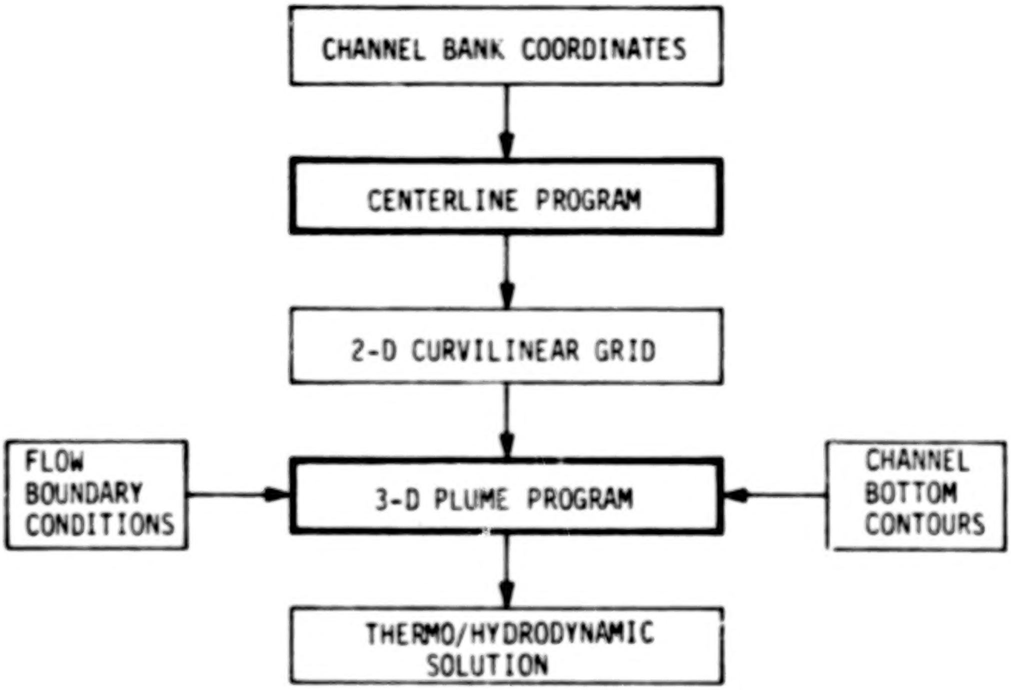

SCALE IN METERS
0
300
600
900
50
31
30
27
23
22
21
20
19
17
7
6
5
3
2
1
1234567

CURVILINEAR GRID FOR TOMBIGBEE RIVER SEGMENT

- PORTION OF TENNESSEE - TOMBIGBEE WATERWAY
- EXTREME SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID

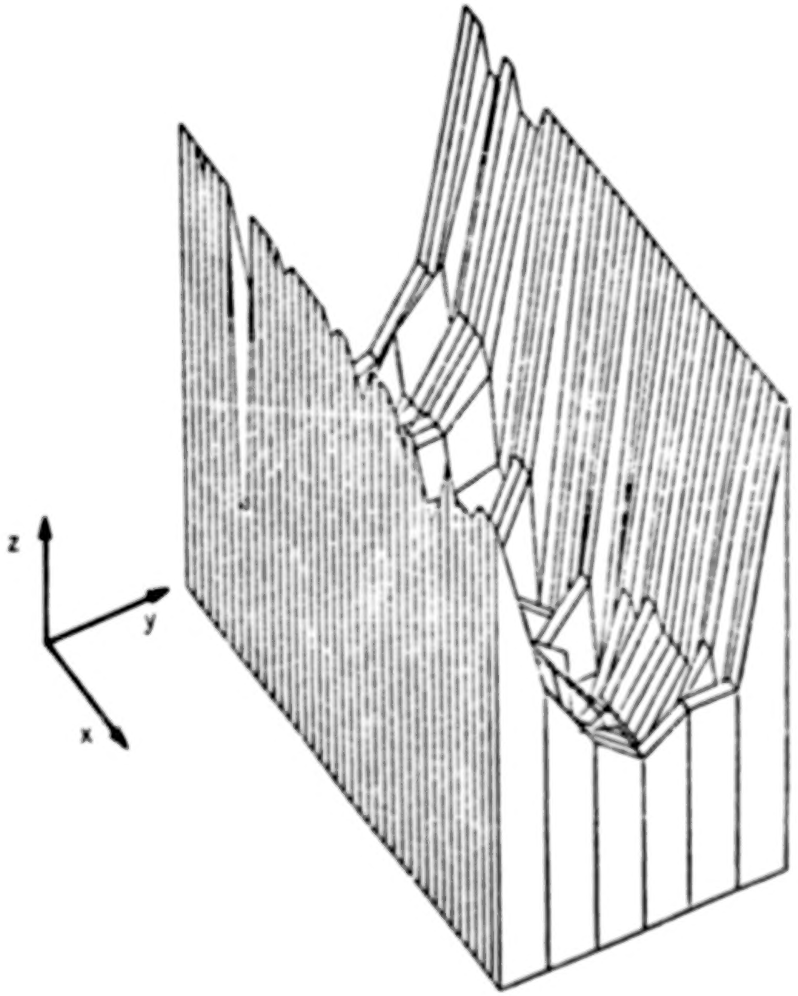

TOMBIGBEE RIVER
PLATE 44
SCALE IN METERS
0
300
600
900

INTEGRATION OF CENTERLINE PROGRAM WITH 3-D PLUME PROGRAM

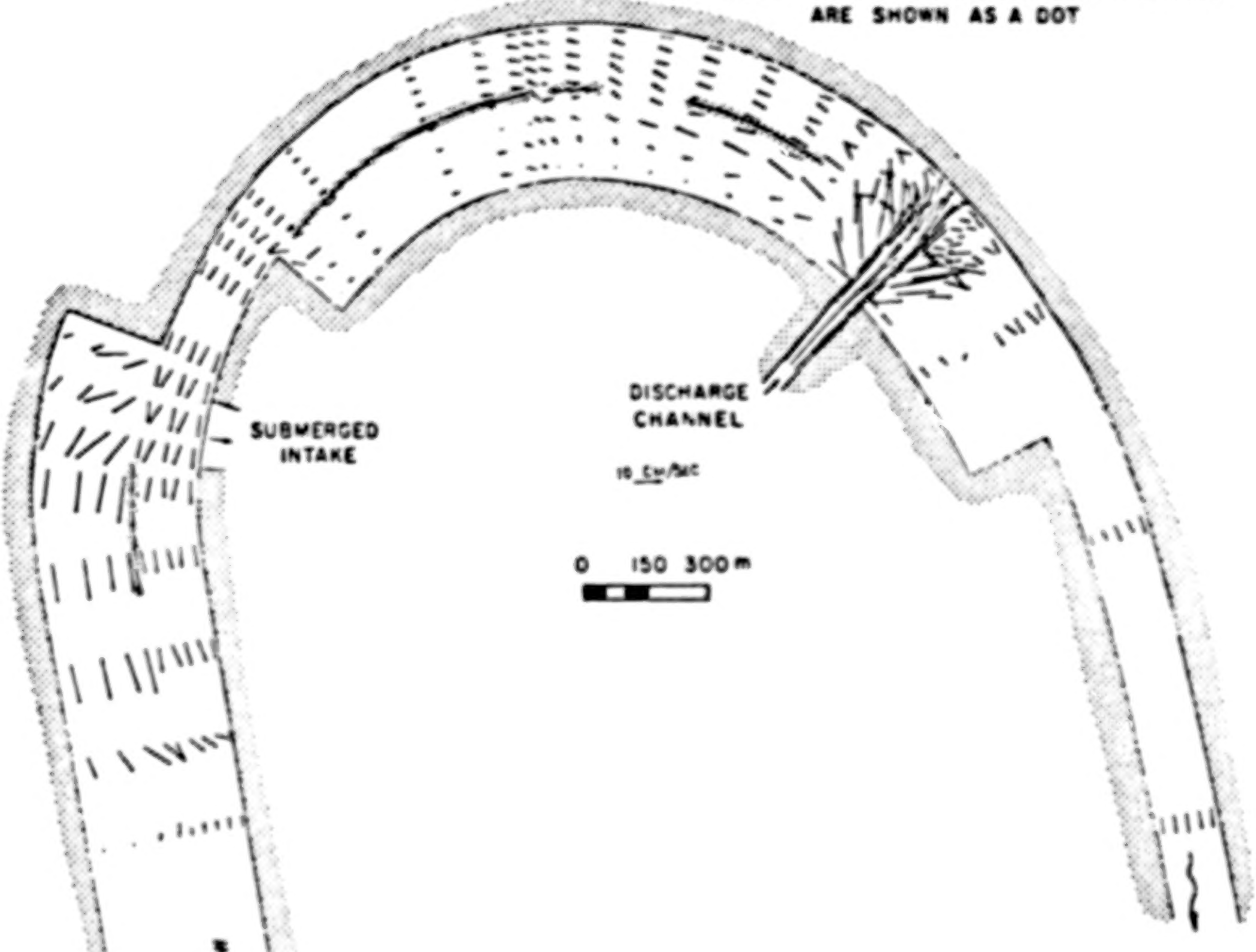

CHANNEL BANK COORDINATES
CENTERLINE PROGRAM
2-D CURVILINEAR GRID
FLOW BOUNDARY CONDITIONS
3-D PLUME PROGRAM
CHANNEL BOTTOM CONTOURS
THERMO/HYDRODYNAMIC SOLUTION

NON-UNIFORM BOTTOM CONSIDERATIONS OF CUMBERLAND RIVER SEGMENT

- BOTTOM PROFILES BASED ON SOUNDINGS
- LONGITUDINAL AND TRANSVERSE VARIATIONS ACCEPTED
- GRID SPACING LIMITS RESOLUTION OF BOTTOM SHAPE
- BOTTOM PROFILES NOT USED FOR TRANSFORMATION

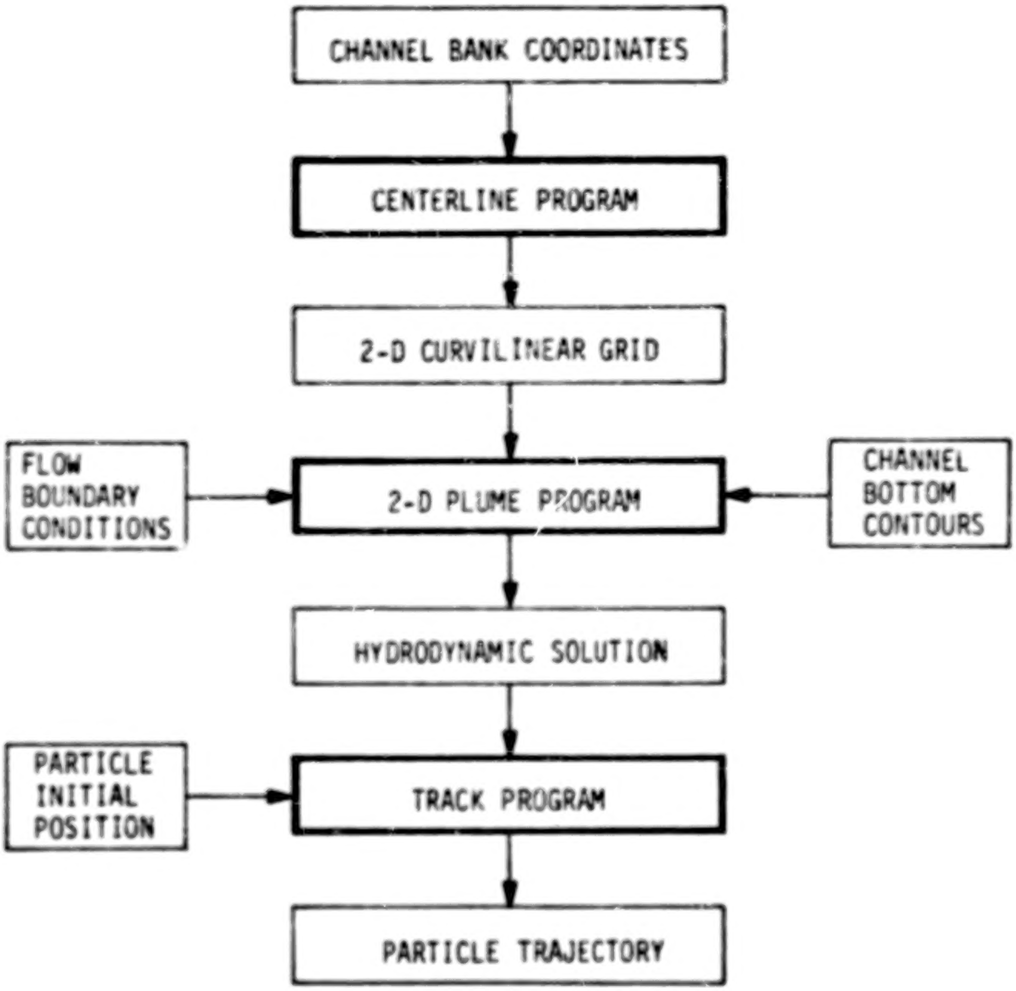


VELOCITY VECTOR PLOT FOR CUMBERLAND RIVER SEGMENT



NOTE: VELOCITIES LESS THAN 1.5 cm/sec ARE SHOWN AS A DOT

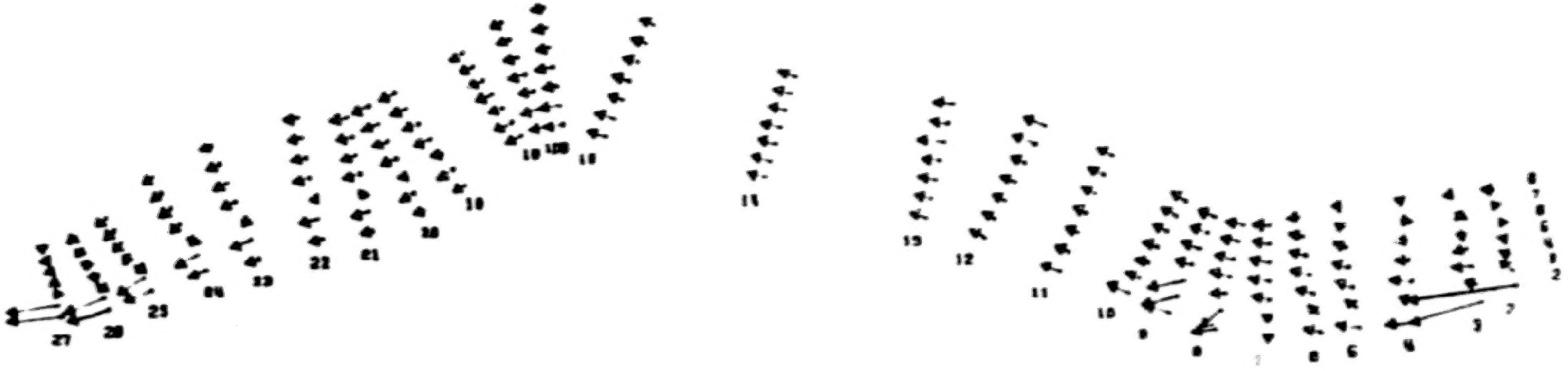
INTEGRATION OF CENTERLINE PROGRAM WITH 2-D PLUME AND TRACK PROGRAMS


CHANNEL BANK COORDINATES
CENTERLINE PROGRAM
2-D CURVILINEAR GRID
FLOW BOUNDARY CONDITIONS
2-D PLUME PROGRAM
CHANNEL BOTTOM CONTOURS
HYDRODYNAMIC SOLUTION
PARTICLE INITIAL POSITION
TRACK PROGRAM
PARTICLE TRAJECTORY

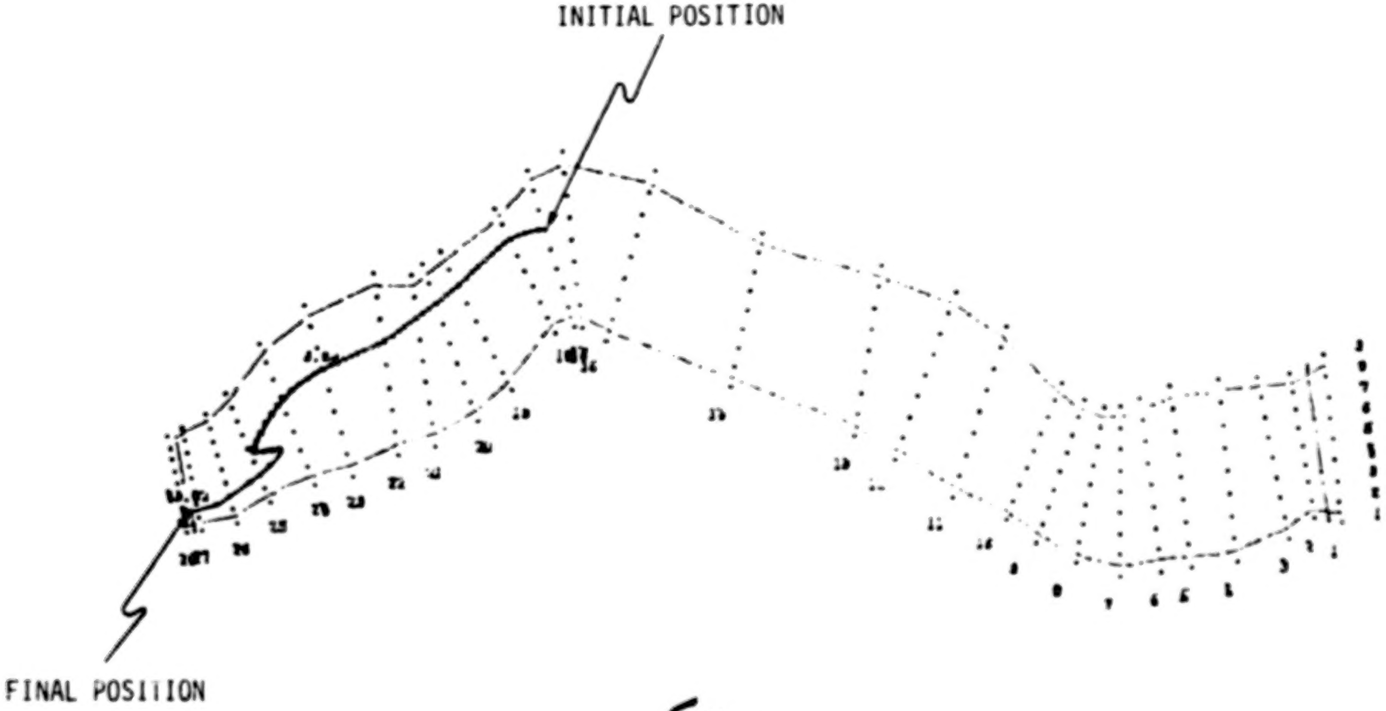
BLANK PAGE

BLANK PAGE

VELOCITY VECTOR PLOT FOR THE TENNESSEE RIVER, WILSON RESERVOIR



PARTICLE TRAJECTORY PLOT FROM TRACK IN TENNESSEE RIVER, WILSON RESERVOIR



CENTERLINE SUMMARY

- APPLICABLE TO SINUOUS RIVER CHANNELS
- CURRENTLY OPERATIONAL
- DIGITIZATION OF CHANNEL COORDINATES
- CONSTANT/VARIABLE CHANNEL WIDTH OPTIONS
- UNIFORM/NON-UNIFORM GRID OPTIONS
- PRESENTLY COUPLED WITH 2-D AND 3-D HYDRODYNAMIC MODELS

BLANK PAGE

BLANK PAGE

GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT FLOWFIELD COMPUTATION

KENNETH E. TATUM
MCDONNELL AIRCRAFT COMPANY
MCDONNELL DOUGLAS CORPORATION
ST. LOUIS, MISSOURI

To facilitate development of the finite element method for fluid dynamics problems a 2-D mesh generation scheme has been developed with the emphasis on versatility and independence of the finite element solution algorithm to be employed. No effort has been expended to maintain grid line orthogonality since the finite element method has no such requirement. The method consists of sequences of shearings and conformal maps with upper and lower surfaces handled independently to allow sharp leading edges. The method will generate meshes of triangular or quadrilateral elements. Thus, with certain additional constraints of smoothness and near-orthogonality, a quadrilateral mesh could be generated for a finite volume type method. Finally, solutions obtained by the MCAIR finite element full potential flow program on sample meshes are shown to illustrate their usefulness.

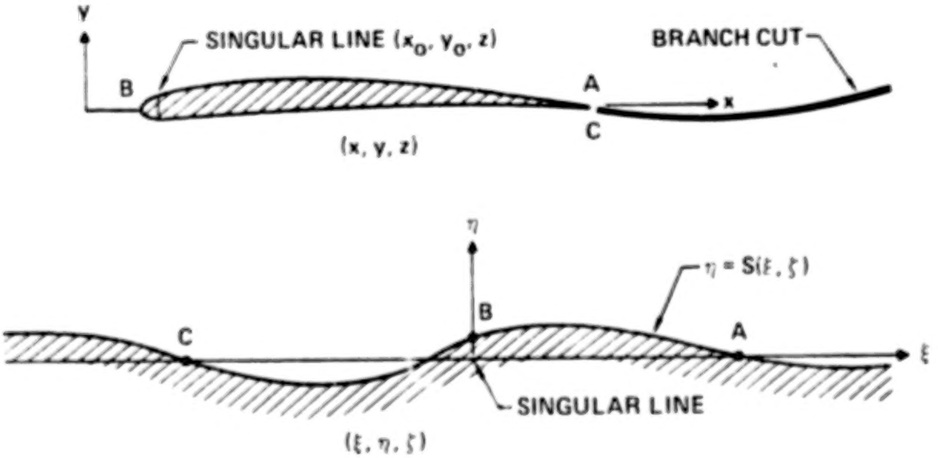
TYPICAL ANALYTIC TRANSFORMATION

PARABOLIC PLUS SHEARING

The parabolic tranformation shown is a typical method used to generate body fitted coordinate meshes for 2-D flowfield computations. Precise transformation Jacobians must be defined relating the uniform cartesian computational grid to the physical body-conforming coordinate grid. Computations are performed by Finite Difference Methods (FDM) in the cartesian coordinate space with determinants of the Jacobians appearing as added coefficients in the difference equations. Simple analytic transformations, even if multiple, cause little increase in complexity of the equations. However for complex body shapes numeric transformation techniques must be employed requiring Jacobian matrices to be computed for each grid cell. These matrices, often approximated, must be stored within the computer or recomputed for successive iterations of nonlinear systems. Either technique is costly.

$$\xi + i\eta = \left\{ [x - x_0(z) + iy - iy_0(z)] / t(z) \right\}^{1/2}$$

$$\zeta = z$$



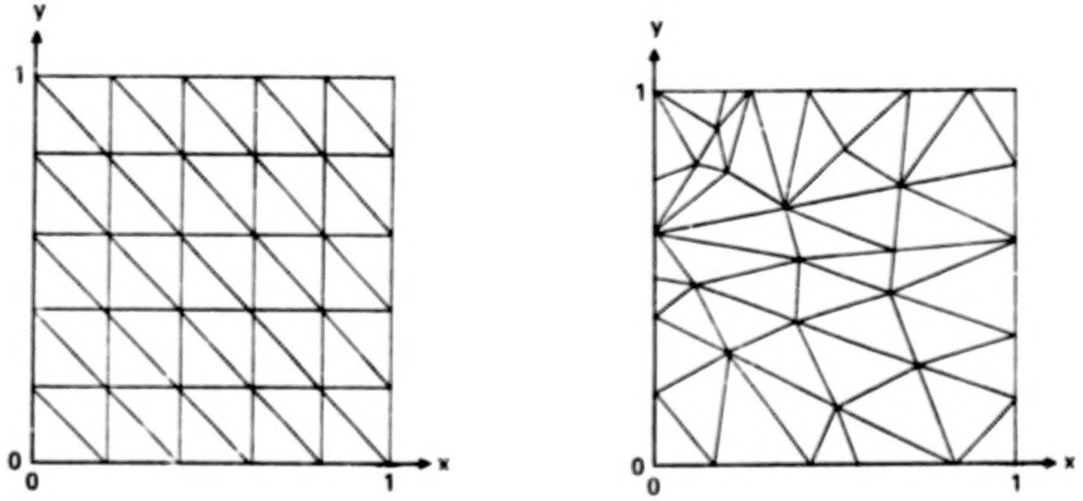
AND THE SHEARING TRANSFORMATION

$$X = \xi \qquad Y = \eta - S(\xi, \zeta) \qquad Z = \zeta$$

Figure 1

POSSIBLE FINITE ELEMENT MESH TRIANGULATION ON UNIT SQUARE

McDonnell Aircraft Company (MCAIR) is studying the Finite Element Method (FEM) as a method which might eliminate, or drastically reduce, the cost associated with transformation Jacobians. The FEM is equally suited to uniform cartesian meshes or irregular, highly non-orthogonal meshes. Two distinctly different FEM meshes (triangulations) of the unit square are shown in Figure 2. Each mesh contains 36 nodes and are equally usable even though a specific problem may indicate the desirability of one over the other. Computations may thus be performed directly in physical space on body-fitted grids generated independently of orthogonality constraints. Only the physical nodal coordinates and the relationships of nodes to elements must be stored.



36 NODES
50 ELEMENTS

36 NODES
48 ELEMENTS

Figure 2

COMPLEX SINE CONFORMAL MAPPING

Grid generation for a FEM computation may be performed by many means. While conformal mappings of simple, highly regular grids are not necessary from the standpoint of maintaining orthogonality, they are useful in producing grids with simple relationships between nodes and elements. Accordingly the current MCAIR technique is based on a conformal (sine) mapping of a rectangular region to a semi-oval region. A sequence of shearing and stretching transformations, both prior to and subsequent to the conformal mapping, shape line $E'F'A'$ to that of one surface of the airfoil, either upper or lower. The number of, or localization of, the shearings is entirely unimportant as long as they may be programmed easily.

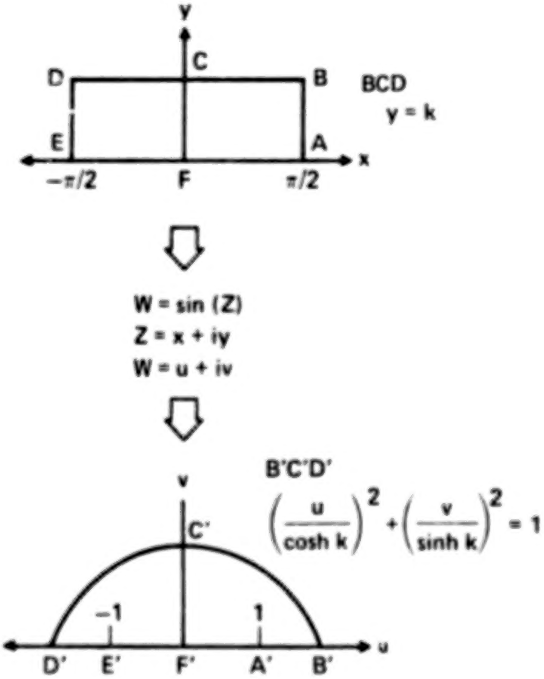

y
C
D
B
BCD
y = k
E
A
x
−π/2
F
π/2
W = sin (Z)
Z = x + iy
W = u + iv
v
B'C'D'
$\left(\frac{u}{\cosh k}\right)^2 + \left(\frac{v}{\sinh k}\right)^2 = 1$
C'
−1
1
u
D'
E'
F'
A'
B'

Figure 3

MATCHING OF UPPER AND LOWER MESH REGIONS

The general method described by Figure 3 is used twice to form two mesh regions as shown in Figure 4, one about the upper surface and one about the lower surface. The airfoil is situated with the forward-most and aft-most points at $y = 0, x = \pm 1$ and the two regions are designed to match along the line $y = 0, |x| \geq 1$. Points along the line AC are doubly specified thus creating a cut across which wake/circulation boundary conditions may be applied. Points along the matching line BD are merged and no boundary is considered there in the final mesh.

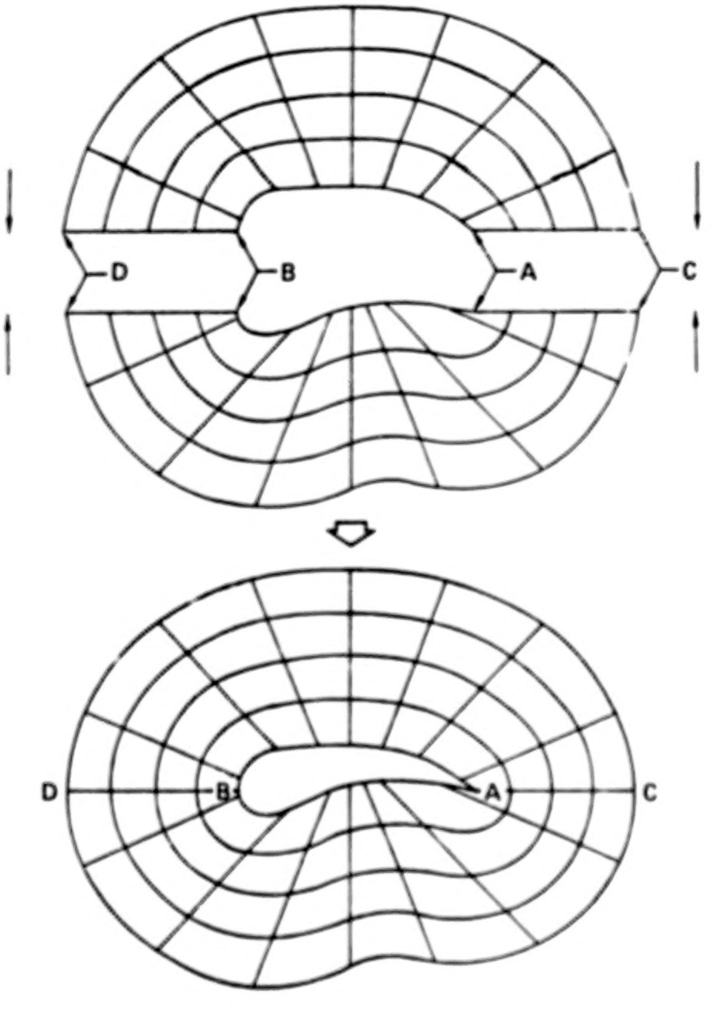


Figure 4

TRIANGULATION OF FIELD BY DIVIDING QUADRILATERALS ALONG APPROPRIATE DIAGONAL

The actual computation of nodal coordinates has been automated in a Fortran computer code for an arbitrary airfoil either specified analytically or by discrete points. Program inputs allow the exact specification of the desired mesh spacing along the body surface as well as the relative spacings normal to the surface. The final field is triangulated as shown in Figure 5 by dividing quadrilaterals along a diagonal, with the diagonal direction varying between regions of the mesh in such a way as to prevent the conformal map from collapsing a triangle to zero area. Triangular elements are desired only because of the simplicity of finite element integration over such regions. However, if desired, quadrilateral elements are obviously generated quite as readily by the scheme.

DASHED LINES = DIAGONAL

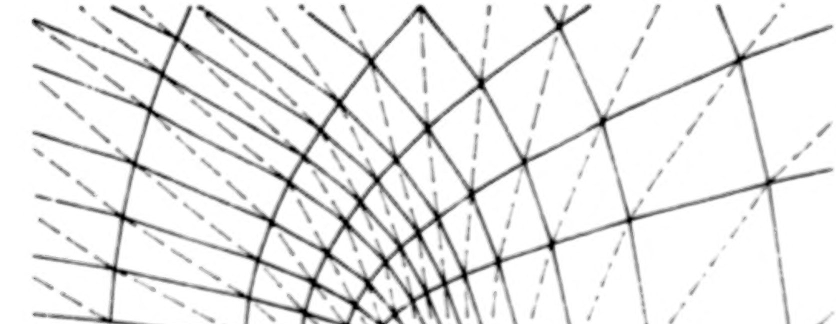


Figure 5

16.3% NLR 7301 AIRFOIL

72 × 17 MESH

Figure 6 shows a final resultant mesh generated about a modern supercritical airfoil, the 16.3% thick NLR 7301 airfoil, with the coordinate system scaled by the airfoil chord. The mesh consists of 72 elements bounding the airfoil surface with 17 rows of elements extending outward from the surface. A total of 1314 nodes define the 2448 elements. Neither the bluntness of the leading edge region nor the reverse curvature of the aft lower surface create any difficulties for the method.

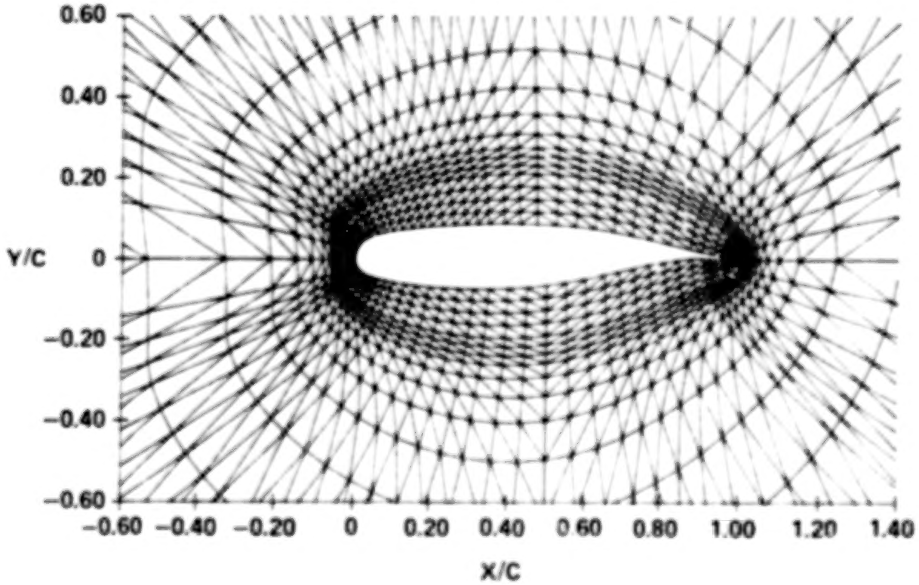

0.60
0.40
0.20
Y/C
0
−0.20
−0.40
−0.60
−0.60
−0.40
−0.20
0
0.20
0.40
0.60
0.80
1.00
1.20
1.40
X/C

Figure 6

6% SYMMETRIC BICONVEX AIRFOIL

72 × 17 MESH

Figure 7 shows a sample mesh about an airfoil with opposite extremes to that of Figure 6. The airfoil is a thin (6%) symmetric Biconvex section with sharp leading and trailing edges. The sharpness of the leading edge presents no difficulties for the method due to the independent handling of upper and lower surfaces. No singular point (x_o, y_o), as needed for example by the parabolic transformation illustrated in Figure 1, exists for this type of leading edge. Thus, any solution procedure depending on a singularity point unwrapping transformation will fail on this airfoil.

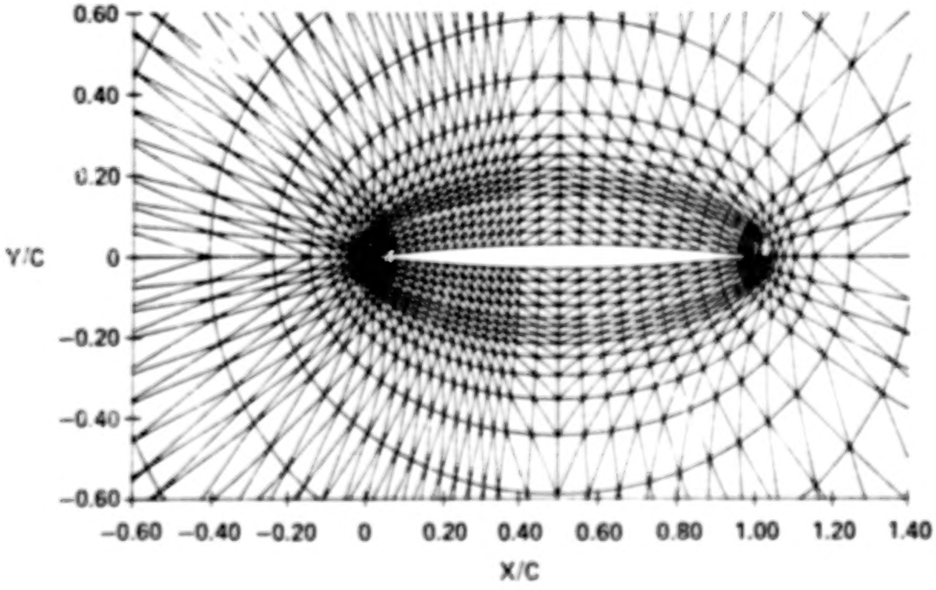


Figure 7

6% SYMMETRIC BICONVEX AIRFOIL
LEADING EDGE

Figure 8 is an enlargement of the leading edge region of the mesh in Figure 7. Some local stretchings of the mesh have been automatically performed by the computer code to prevent some elements from becoming too small or thin. The stretchings may distort the smooth variation of elements around the leading edge but in reality increase the potential for obtaining an accurate finite element solution on the mesh due to the maintaining of smaller aspect ratios (maximum/minimum dimension of triangle) of individual elements. Another constraint on the elements necessitating some local stretching is that the magnitude of the area of the smallest element not be too small relative to significant digit resolution of the computer on which calculations are to be performed.

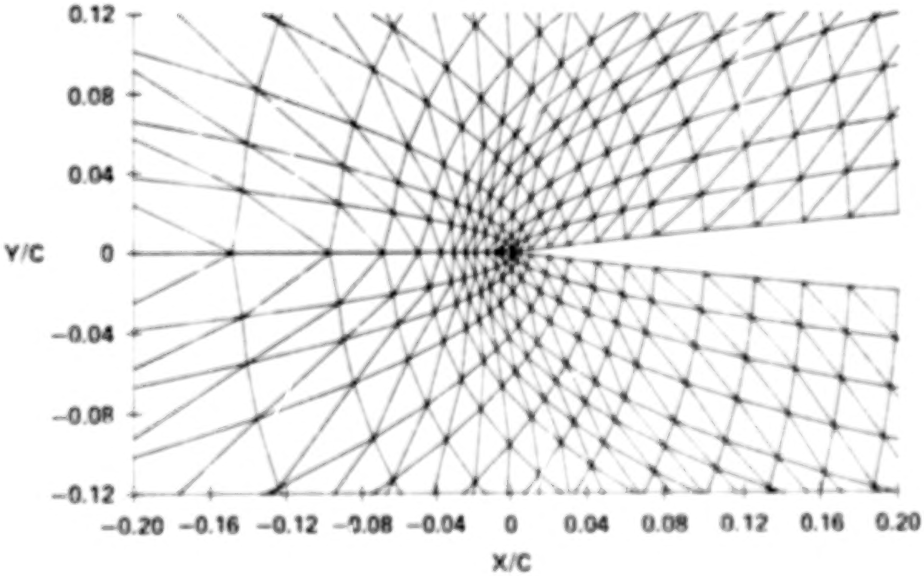


Figure 8

SUPERSONIC FIGHTER AIRFOIL (20° AND 10° FLAPS)

72 × 17 MESH

The use of shearing transformations not restricted to maintaining orthogonality of the grid allows the creation of grids about sharp corners. Figure 9 shows a grid about a supersonic fighter airfoil section with both leading and trailing edge flaps deflected. The discontinuous surface slopes at the hinge lines might create numerical singularities in methods which attempt to maintain orthogonal grids. The MCAIR technique however has no such difficulties. The computer program also allows element spacings to be user specified to facilitate bunching of elements around the flap hinges where high velocity gradients are to be expected in the FEM solution.

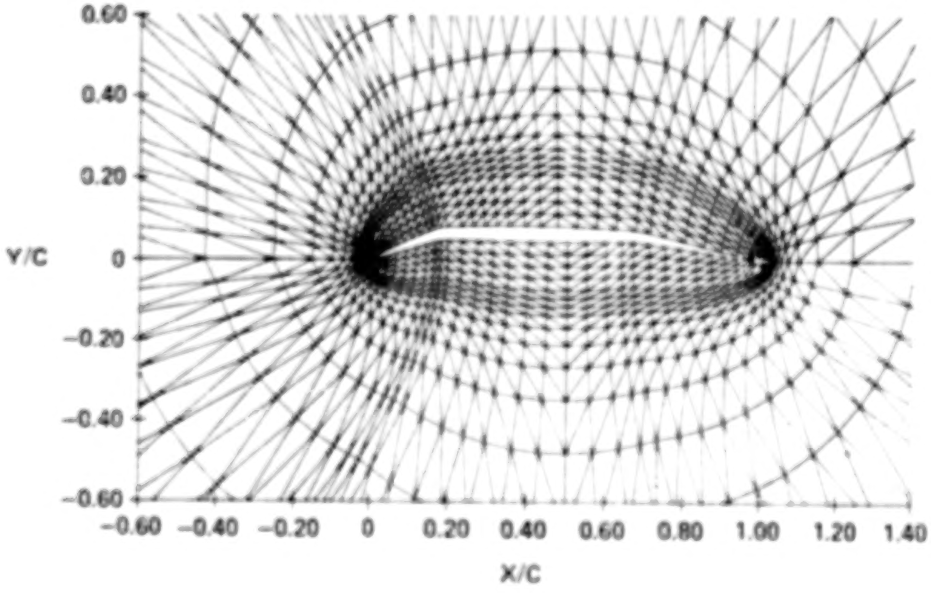


Figure 9

NLR SYMMETRIC SHOCK-FREE AIRFOIL

36 × 17 MESH

Figure 10 is an example of a half-plane mesh about a symmetric NLR shock-free airfoil design. The independent handling of upper and lower airfoil surfaces allows this type of mesh to be generated very simply. Flow solutions about symmetric airfoils at zero angle-of-attack may thus be obtained at half the expense or with double the nodal density but no additional cost. Such solutions are important in fundamental research and also for comparison of full potential flow solutions with small perturbation solutions. These small perturbation solutions are most strictly valid at small, or zero, angle-of-attack on thin or sharp nosed airfoils such as the biconvex section shown in Figure 7.

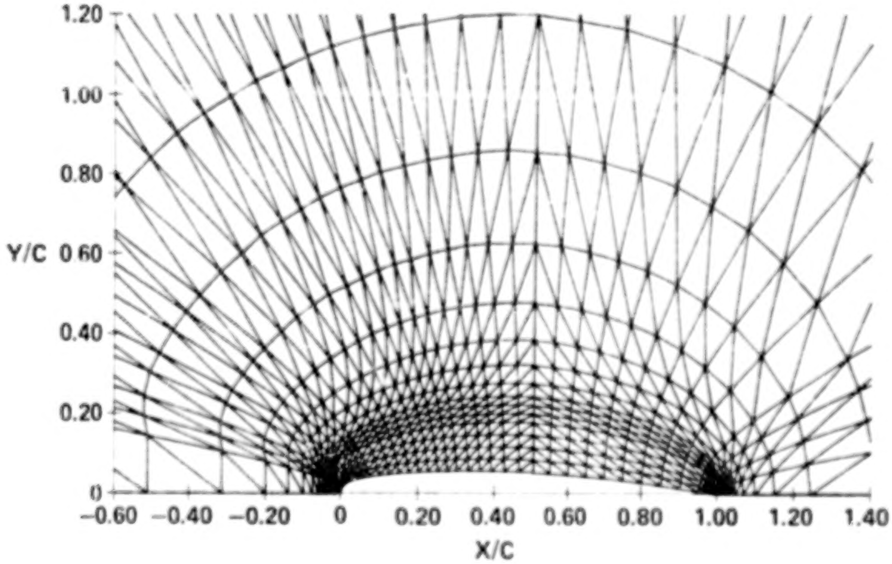


Figure 10

16.3% NLR 7301 AIRFOIL

ALPHA = 0.391° MACH 0.502

Figures 11-13 illustrate the use of the meshes shown in previous figures by the current MCAIR FEM full potential flow program. Figure 11 compares a FEM solution at a moderate subsonic Mach number (0.5) and small angle-of-attack to the solution by a modern state-of-the-art Finite Difference Method (FDM) program. The comparison is good even though the FDM grid was much denser than the FEM mesh.

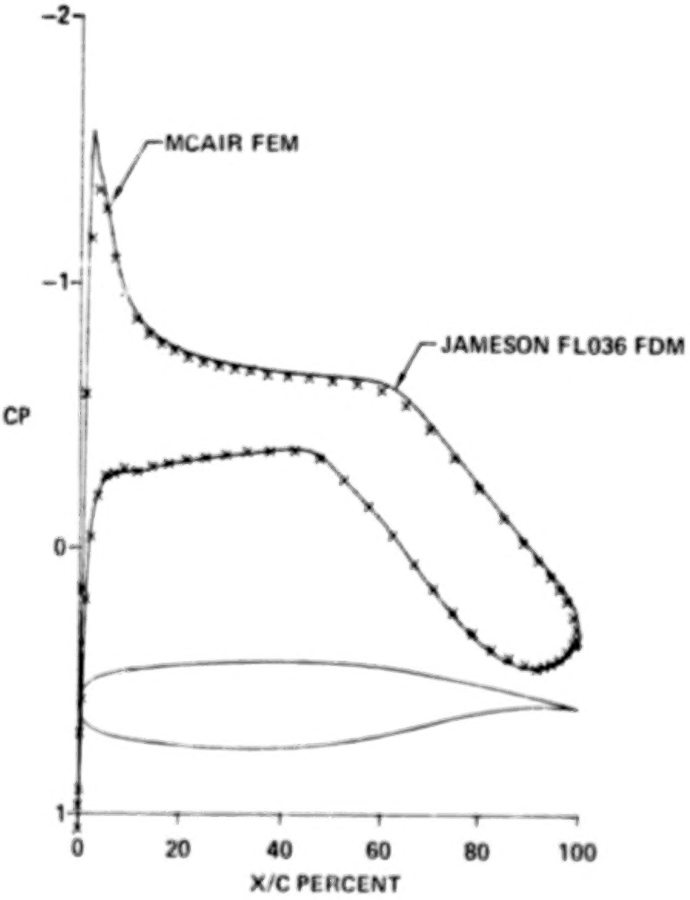


Figure 11

SUPERSONIC FIGHTER AIRFOIL (20° AND 10° FLAPS)

ALPHA = 1.0° MACH 0.5

Figure 12 compares solutions for the supersonic fighter airfoil with 20° leading edge and 10° trailing edge flaps. Since no finite difference program was available which would compute flows about sharp leading edges and abrupt hinge lines, a modern technology Panel Method program, the Bristow Multielement Airfoil Analysis and Design (MAAD) program, was employed for comparison. A Karman-Tsien compressibility correction was applied to obtain a Mach 0.5 solution. Agreement is good even in regions of pressure spikes; however, the inexpensive panel method employed approximately double the panel (solution node) density of the FEM and was able to more accurately resolve the pressure peaks.

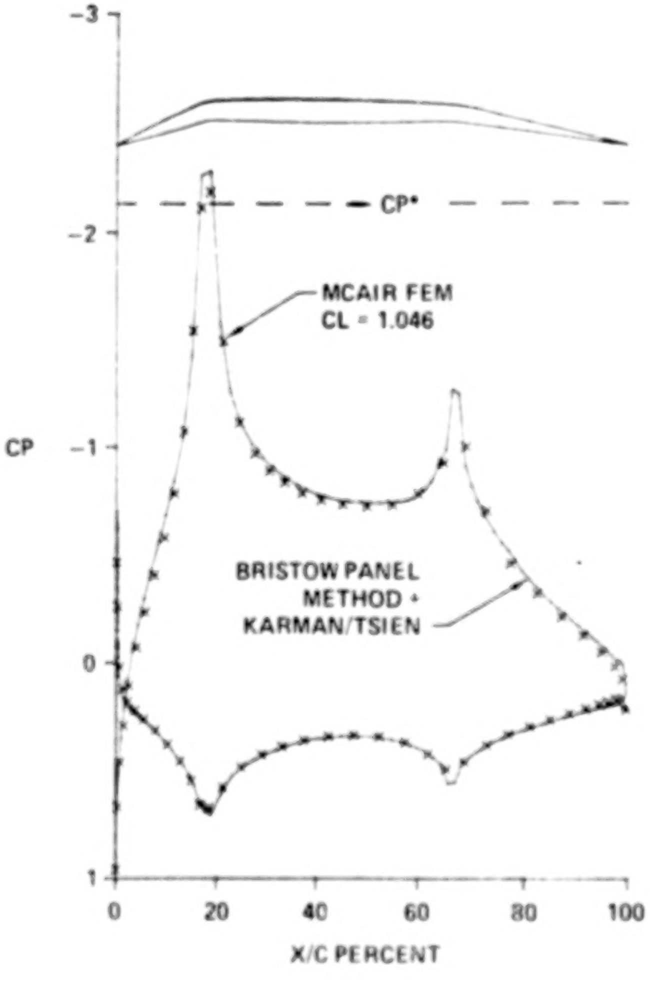


Figure 12

NLR SYMMETRIC AIRFOIL

ALPHA = 0° MACH 0.786

Figure 13 illustrates the use of a half-plane mesh, specifically that shown in Figure 9. The solution was obtained in the fundamental research on adapting the FEM to non-subsonic flowfields where the governing differential equations are of mixed elliptic/hyperbolic type. Using the artificial density concept of Holst the MCAIR FEM was able to produce this solution on a shock-free airfoil which agrees reasonably well with the theoretical hodograph solution. The versatility of the mesh generator, both in technique and program, greatly facilitates the FEM research into fundamental computational methods and in applied fluid mechanics.

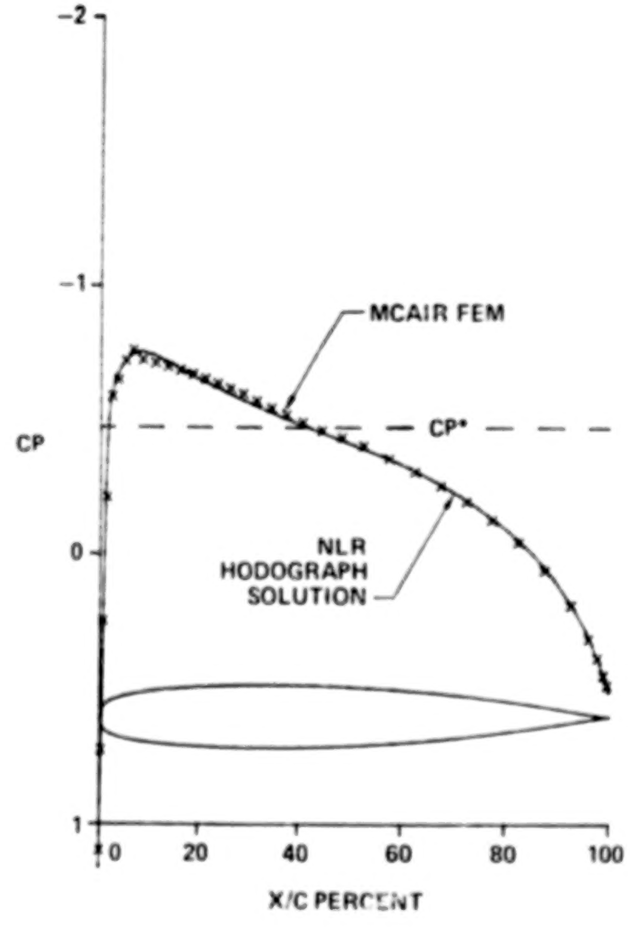


Figure 13

Numerical Generation of Two-Dimensional Orthogonal Curvilinear Coordinates in an Euclidean Space*

Z. U. A. Warsi and J. F. Thompson
Department of Aerospace Engineering
Mississippi State University
Mississippi State, MS 39762

Abstract

In this paper a non-iterative method for the numerical generation of orthogonal curvilinear coordinates for plane annular regions between two arbitrary smooth closed curves has been developed. The basic generating equation is the Gaussian equation for an Euclidean space which has been solved analytically. The method has been applied in many cases and these test results demonstrate that the proposed method can be readily applied to a wide variety of problems. The method can also be used for simply connected regions only by obtaining the solution of the linear equation (19) under the changed boundary conditions. Details on the work reported in this paper are available in Reference [1].

---

*This work has been supported in part by the Air Force Office of Scientific Research, under Grant AFOSR No. 76-2922 and AFOSR No. 80-0185.

Fundamental Ideas of the Method

All methods of numerical coordinate generation in a two-dimensional plane and classified under the method of "elliptic equations" (Refs. [2]-[10]), have depended invariably on the solution of Poisson equations for the curvilinear coordinates $\xi(x,y)$ and $\eta(x,y)$:

$$\nabla^2 \xi = -\frac{1}{g} (g_{11}\Gamma^1_{22} - 2g_{12}\Gamma^1_{12} + g_{22}\Gamma^1_{11})$$

$$= -\frac{g_{22}}{g} P(\xi,\eta) \tag{1a}$$

$$\nabla^2 \eta = -\frac{1}{g} (g_{11}\Gamma^2_{22} - 2g_{12}\Gamma^2_{12} + g_{22}\Gamma^2_{11})$$

$$= -\frac{g_{11}}{g} Q(\xi,\eta) \tag{1b}$$

where $P(\xi,\eta)$, $Q(\xi,\eta)$ are arbitrarily specified control functions, the g_{ij} are the fundamental metric coefficients, the Γ^i_{jk} are the Christoffel symbols of the second kind

$$\Gamma^i_{jk} = g^{i\ell}[jk,\ell] \tag{2}$$

$$[jk,\ell] = \frac{1}{2} \left(\frac{\partial g_{j\ell}}{\partial x^k} + \frac{\partial g_{k\ell}}{\partial x^j} - \frac{\partial g_{jk}}{\partial x^\ell}\right)$$

and

$$g = g_{11}g_{22} - (g_{12})^2$$

Implicitly equation (1) implies two things: (i) that the coordinates for the same domain can also be obtained by solving the Laplace equations

$$\nabla^2 \xi = 0, \quad \nabla^2 \eta = 0 \tag{3}$$

and (ii) since the Γ^i_{jk} have first partial derivatives of g_{ij} in them, equation (1) can also be interpreted as providing a set of constraints or relations among the g_{ij} and their first partial derivatives.

In this paper we present another method based on elliptic equations and state the problem as follows.

The three functions g_{11}, g_{12}, g_{22} of the curvilinear coordinates ξ,η define an element of length ds in a plane if the Gaussian equation with zero curvature

$$\frac{\partial}{\partial\eta}\left(\frac{\sqrt{g}\,\Gamma^2_{11}}{g_{11}}\right) - \frac{\partial}{\partial\xi}\left(\frac{\sqrt{g}\,\Gamma^2_{12}}{g_{11}}\right) = 0 \qquad (4)$$

holds for every point in the plane, and then the Cartesian coordinates are given as

$$x = x(\xi,\eta), \quad y = y(\xi,\eta)$$

Equation (4) is identically satisfied by a function $\alpha(\xi,\eta)$ defined as

$$\alpha_\xi = \frac{-\sqrt{g}}{g_{11}}\Gamma^2_{11}, \quad \alpha_\eta = \frac{-\sqrt{g}}{g_{11}}\Gamma^2_{12}$$

Specifically, α is the angle of inclination with respect to the x-axis of the tangent to the coordinate line η = const. directed in the sense of increasing values of the parameter ξ. The first partial derivatives of x and y are

$$\left.\begin{aligned} x_\xi &= \sqrt{g_{11}}\cos\alpha, \quad y_\xi = -\sqrt{g_{11}}\sin\alpha \\ x_\eta &= \frac{1}{\sqrt{g_{11}}}(g_{12}\cos\alpha + \sqrt{g}\sin\alpha), \quad y_\eta = -\frac{1}{\sqrt{g_{11}}}(\sqrt{g}\cos\alpha - g_{12}\sin\alpha) \end{aligned}\right\} \qquad (5)$$

Then

$$\left.\begin{aligned} x &= \int\left[\sqrt{g_{11}}\cos\alpha\, d\xi + \frac{1}{\sqrt{g_{11}}}(g_{12}\cos\alpha + \sqrt{g}\sin\alpha)d\eta\right] \\ y &= -\int\left[\sqrt{g_{11}}\sin\alpha\, d\xi - \frac{1}{\sqrt{g_{11}}}(\sqrt{g}\cos\alpha - g_{12}\sin\alpha)d\eta\right] \\ \alpha &= -\int \frac{\sqrt{g}}{g_{11}}(\Gamma^2_{11}\, d\xi + \Gamma^2_{12}\, d\eta) \end{aligned}\right\} \tag{6}$$

The inverse relations of (5) are

$$\left.\begin{aligned} \xi_x &= (\sqrt{g}\cos\alpha - g_{12}\sin\alpha)/\sqrt{gg_{11}} \\ \xi_y &= -(g_{12}\cos\alpha + \sqrt{g}\sin\alpha)/\sqrt{gg_{11}} \\ \eta_x &= \sqrt{g_{11}/g}\,\sin\alpha \\ \eta_y &= \sqrt{g_{11}/g}\,\cos\alpha \end{aligned}\right\} \tag{7}$$

For the case of orthogonal coordinates, the coefficient $g_{12} = 0$, i.e.,

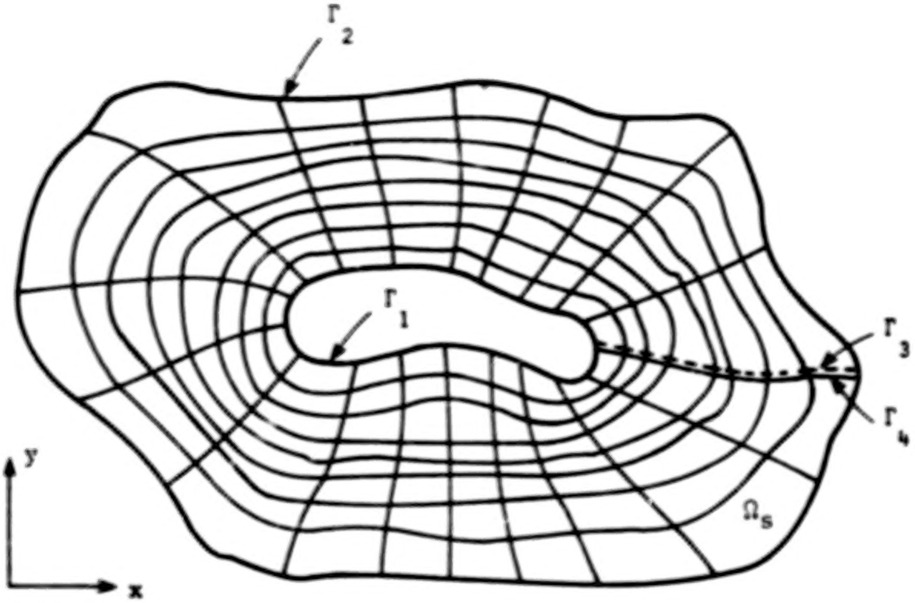
$$g_{12} = x_\xi x_\eta + y_\xi y_\eta = 0 \tag{8}$$

which is satisfied by the equations

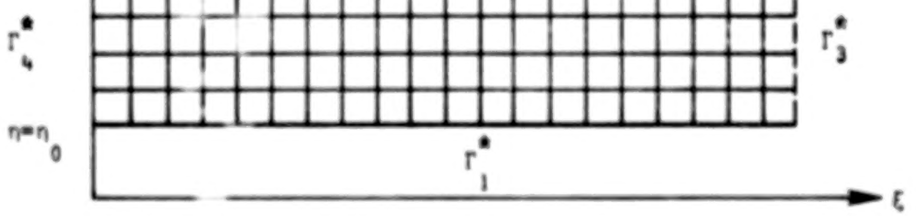
$$\left.\begin{aligned} x_\eta &= -Fy_\xi \\ y_\eta &= Fx_\xi \end{aligned}\right\} \tag{9}$$

where $F > 0$ is a continuous function of g_{11} and g_{22} [11] .

Referring to Figure 1, let the boundary Γ_2 of a bounded region in an Euclidean two-dimensional space be a simple curve $x = x_\infty(\xi)$, $y = y_\infty(\xi)$, with a uniformaly turning tangent. In the region Ω, let



Γ_2
Γ_1
Γ_3
Γ_4
Ω_s
y
x



η
Γ_2^*
$\eta = \eta_\infty$
Γ_4^*
Γ_3^*
$\eta = \eta_0$
Γ_1^*
ξ

Transformed Plane
(Natural Coordinates)

Figure 1.- Physical and transformed planes.

Ω_s be an annular subregion bounded by the inner boundary Γ_1 and the outer boundary Γ_2. The region Ω_s is to be mapped onto a rectangular region R in the $\xi\eta$-plane by a transformation so as to have

$$\left.\begin{aligned} x &= x(\xi,\eta) \\ y &= y(\xi,\eta) \end{aligned}\right\} \qquad \eta_\beta \leq \eta \leq \eta_\infty$$

where η_β and η_∞ are the actual parametric values associated with the boundaries Γ_1 and Γ_2, respectively, and x,y are periodic in the ξ-argument.

Substituting $g_{12} = 0$ in the fundamental equation (4) we get

$$\frac{\partial}{\partial\xi}\left[\frac{1}{Fg_{11}}\frac{\partial}{\partial\xi}(F^2 g_{11})\right] + \frac{\partial}{\partial\eta}\left[\frac{1}{Fg_{11}}\frac{\partial g_{11}}{\partial\eta}\right] = 0 \tag{10}$$

where

$$\left.\begin{aligned} g_{22} &= F^2 g_{11} \\ g &= (Fg_{11})^2 \end{aligned}\right\} \tag{11}$$

Before we solve the problem of orthogonal coordinate generation based on the elliptic equation (10), we digress and state the following results: Following Potter and Tuttle [6] we assume that the ξ-curves in the physical xy-plane are free from sources and sinks. This condition establishes a unique correspondence between the ξ-points on each pair of $\eta =$ const. lines. In the absence of sources and sinks, we have

$$\operatorname{div}[\operatorname{grad}\psi(\eta)] = 0 \tag{12}$$

where $\psi(\eta)$ is an arbitrary differentiable function of η and grad $\psi(\eta)$ is oriented along the normal to the curve $\eta =$ const. Carrying out the differential operation in (12) and using the expressions

$$|\operatorname{grad}\eta| = 1/\sqrt{g_{22}}$$

$$\nabla^2\eta = \frac{1}{\sqrt{g}}\frac{\partial}{\partial\eta}\left(\sqrt{\frac{g_{11}}{g_{22}}}\right)$$

in (12), we obtain

$$\frac{\partial}{\partial\eta}\left(\ln\sqrt{g_{11}/g_{22}}\right) = -\frac{d^2\psi}{d\eta^2}\Big/\frac{d\psi}{d\eta}$$

Writing $\frac{d\psi}{d\eta} = 1/\nu(\eta)$ and denoting the arbitrary function of ξ due to integration by $\ln\mu(\xi)$, we obtain the result

$$\begin{aligned}\sqrt{g_{11}/g_{22}} &= \mu(\xi)\nu(\eta)\\ &= 1/F \qquad (13)\end{aligned}$$

This result shows that for the case of orthogonal coordinates the ratio g_{11}/g_{22} is a product of the positive functions $\mu(\xi)$ and $\nu(\eta)$. The result in (13) also provides the condition for the two distinct families of orthogonal curves

$$\xi = \text{const.},\ \eta = \text{const.}$$

to divide the physical plane in infinitensimal squares. (See Cohen [12]).

We now introduce new coordinates $\xi'(\xi)$ and $\eta'(\eta)$ as

$$\xi' = \int\mu(\xi)d\xi,\ \eta' = \int\frac{d\eta}{\nu(\eta)} \qquad (14)$$

Thus

$$g'_{11} = g_{11}/\mu^2,\ g'_{22} = g_{22}\nu^2$$

so that

$$g'_{22} = g'_{11}$$

Defining

$$P' = \ln g'_{11}$$

it can be shown that

$$\frac{\partial^2 P'}{\partial \xi'^2} + \frac{\partial^2 P'}{\partial \eta'^2} = \frac{\nu(\eta)}{\mu(\xi)} \{\frac{\partial}{\partial \xi} [\frac{1}{F g_{11}} \frac{\partial}{\partial \xi} (F^2 g_{11})] + \frac{\partial}{\partial \eta} [\frac{1}{F g_{11}} \frac{\partial g_{11}}{\partial \eta}]\} \tag{15}$$

Using (15) in (10), we get a much simpler equation

$$\frac{\partial^2 P'}{\partial \xi'^2} + \frac{\partial^2 P'}{\partial \eta'^2} = 0 \tag{16}$$

Another important result can be obtained based on (13). Using the orthogonality condition $g_{12} = 0$ in (7), we have

$$\eta_y = \xi_x/F, \ \eta_x = - \xi_y/F$$

so that

$$\frac{\partial}{\partial \xi} (\xi_x/F) + \frac{\partial}{\partial y} (\xi_y/F) = 0 \tag{17}$$

Carrying out the transformation (14) in (17), we get

$$\nabla^2 \xi' = 0 \tag{18}$$

Equation (18) provides the uniqueness condition for the solution of equation (16).

Based on the preceeding analysis we can state that if an exact analytic solution of equation (10) can be obtained for $F = 1$, i.e., $g_{22} = g_{11}$, then the solution for any other coordinate system $\bar{\xi}$ and $\bar{\eta}$, where $\xi = \phi(\bar{\xi})$ and $\eta = f(\bar{\eta})$, can simply be obtained by the substitution of ϕ and f in place ξ and η respectively. With this scheme in mind, we solve the equation

$$\frac{\partial^2 P}{\partial \xi^2} + \frac{\partial^2 P}{\partial \eta^2} = 0 \tag{19}$$

where

$$g_{22} = g_{11}$$

$$P = \ln g_{11} \tag{20}$$

under the boundary conditions

$$\left.\begin{aligned} P &= P_\beta(\xi) \text{ at } \eta = 0^\dagger \\ &= P_\infty(\xi) \text{ at } \eta = \eta_\infty \end{aligned}\right\} \quad 0 \le \xi \le 2\pi \tag{21}$$

where the subscripts β and ∞ denote the inner and outer boundaries, respectively. The periodicity requirement is that

$$P(\xi,\eta) = P(\xi+2\pi,\eta) \tag{22}$$

Further, the ξ-coordinate must be such that the equation

$$\nabla^2\xi = 0 \tag{23}$$

is always satisfied.

A general analytic solution of equation (19) under the boundary conditions (21) and the periodicity condition (22) is

$$\begin{aligned} P(\xi,\eta) = a_o + \eta\overline{K} + \sum_{n=1}^{\infty} \sinh n(\eta_\infty-\eta) \\ (a_n \cos n\xi + b_n \sin n\xi)/\sinh n\eta_\infty \\ + \sum_{n=1}^{\infty} \sinh n\eta(c_n \cos n\xi + d_n \sin n\xi)/\sinh n\eta_\infty \end{aligned} \tag{24}$$

where

$$\overline{K} = (c_o-a_o)/\eta_\infty \tag{25}$$

and

$$a_o = \frac{1}{2\pi}\int_o^{2\pi} P_\beta(\xi)d\xi, \quad c_o = \frac{1}{2\pi}\int_o^{2\pi} P_\beta(\xi)d\xi$$

[†] There is no loss of generality in setting the parametric value $\eta_\beta = 0$. The value η_∞ must be interpreted as the difference between the actual values of η at the outer and inner boundaries.

$$\left.\begin{aligned} a_n &= \frac{1}{\pi}\int_0^{2\pi} P_\beta(\xi)\cos n\xi\, d\xi, \quad b_n = \frac{1}{\pi}\int_0^{2\pi} P_\beta(\xi)\sin n\xi\, d\xi \\ c_n &= \frac{1}{\pi}\int_0^{2\pi} P_\infty(\xi)\cos n\xi\, d\xi, \quad d_n = \frac{1}{\pi}\int_0^{2\pi} P_\infty(\xi)\sin n\xi\, d\xi \end{aligned}\right\} \quad (26)$$

For orthogonal Coordinates

$$\alpha_\xi = \frac{1}{2\sqrt{g}}\frac{\partial g_{11}}{\partial\eta}, \quad \alpha_\eta = -\frac{1}{2\sqrt{g}}\frac{\partial g_{22}}{\partial\xi}$$

therefore for $g_{22} = g_{11}$

$$\alpha_\xi = \frac{1}{2}\frac{\partial P}{\partial\eta}, \quad \alpha_\eta = -\frac{1}{2}\frac{\partial P}{\partial\xi}$$

and consequently

$$\alpha(\xi,\eta) = \alpha(\xi,o) + \sum_{n=1}^{\infty}\frac{\cosh n(\eta_\infty-\eta)}{2\sinh n\eta_\infty}(b_n\cos n\xi - a_n\sin n\xi)$$

$$+ \sum_{n=1}^{\infty}\frac{\cosh n\eta}{2\sinh n\eta_\infty}(c_n\sin n\xi - d_n\cos n\xi)$$

$$- \sum_{n=1}^{\infty}\frac{\cosh n\eta_\infty}{2\sinh n\nu_\infty}(b_n\cos n\xi - a_n\sin n\xi)$$

$$- \sum_{n=1}^{\infty}\frac{1}{2\sinh n\eta_\infty}(c_n\sin n\xi - d_n\cos n\xi)$$

Having determined g_{11}, and α, we can find the Cartesian coordinates

$$\left.\begin{aligned} x(\xi,\eta) &= x(\xi,o) + \int_o^\eta \sqrt{g_{22}}\sin\alpha\, d\eta \\ y(\xi,\eta) &= y(\xi,o) + \int_o^\eta \sqrt{g_{22}}\cos\alpha\, d\eta \end{aligned}\right\} \quad (28)$$

The preceding solution is for the case when $g_{22} = g_{11}$, i.e., $F = 1$. However, as stated earlier, the solution for any other coordinate system $\bar{\xi},\bar{\eta}$ in which $\bar{g}_{22} \neq \bar{g}_{11}$ can be obtained by replacing ξ and η in (24), (27) and (28) by $\phi(\bar{\xi})$ and $f(\bar{\eta})$, respectively. This feature can be used to redistribute the coordinate lines in the desired regions. Since the functions ϕ and f are at our disposal, they play the same role as P and Q in equation (1). Further, since the Fourier coefficients (26) are invariant to a coordinate transformation $\xi = \phi(\bar{\xi})$, where $\phi(\bar{\xi}_o) = 0$, $\phi(\bar{\xi}_m) = 2\pi$ and $\bar{\xi}_o$ corresponds to $\xi = 0$, $\bar{\xi}_m$ corresponds to $\xi = 2\pi$, these coefficients need not be recalculated.

The procedure of transformation from ξ,η to $\bar{\xi},\bar{\eta}$ is as follows.

On transformation from (ξ,η) to $(\bar{\xi},\bar{\eta})$, the covariant metric coefficients transform through the equation

$$\bar{g}_{ij} = g_{k\ell} \frac{\partial x^k}{\partial \bar{x}^i} \frac{\partial x^\ell}{\partial \bar{x}^j}$$

so that, on using the relations $g_{22} = g_{11}$ and $g_{12} = 0$, we have

$$\left.\begin{aligned} \bar{g}_{11} &= \left[\left(\frac{\partial \xi}{\partial \bar{\xi}}\right)^2 + \left(\frac{\partial \eta}{\partial \bar{\xi}}\right)^2\right] g_{11} \\ \bar{g}_{22} &= \left[\left(\frac{\partial \xi}{\partial \bar{\eta}}\right)^2 + \left(\frac{\partial \eta}{\partial \bar{\eta}}\right)^2\right] g_{11} \end{aligned}\right\} \qquad (29)$$

We now introduce the transformation

$$\left.\begin{aligned} \xi &= \phi(\bar{\xi}) \\ \eta &= f(\bar{\eta}) \end{aligned}\right\} \qquad (30)$$

where the functions ϕ and f are continuously differentiable and satisfy the conditions

$$\phi(\bar{\xi}_o) = 0, \quad f(\bar{\eta}_\beta) = 0$$

where $\xi = 0$ corresponds to $\bar{\xi} = \bar{\xi}_o$ and $\eta = 0$ corresponds to $\bar{\eta} = \bar{\eta}_\beta$.

Defining

$$\lambda = \frac{d\phi}{d\bar{\xi}}, \quad \theta = \frac{df}{d\bar{\eta}}$$

we obtain from (30)

$$\bar{g}_{22}(\bar{\xi},\bar{\eta}) = \frac{\theta^2}{\lambda^2} \; \bar{g}_{11}(\bar{\xi},\bar{\eta})$$

Comparing with (13), we find $\mu = \lambda$, $\nu = \frac{1}{\theta}$. (31)

The salient feature of the preceding analysis is that the solution for the case $g_{22} = g_{11}$ can be used to obtain the solution for the case $g_{22} \neq g_{11}$ by coordinate transformation.

Before solving any specific problem, it is important first to establish an orthogonal correspondence between unique points of the inner and outer boundary curves. This condition is satisfied if we choose the ξ-curves satisfying the equation

$$\nabla^2 \xi = 0 \tag{32}$$

The inner and outer boundary curves are available either in tabular or functional form as

$$y_\beta = y(x_\beta), \; y_\infty = y(x_\infty) \tag{33}$$

For equation (32) to be satisfied, we can take ξ as the angle traced out in a clockwise sense by the common radius of the concentric circles in a conformal representation of the inner and outer boundary curves. If a and A, respectively, are the radii of the inner and outer circles in the transformed conformal plane, then

$$\left.\begin{aligned} a &= \frac{1}{2\pi} \int_0^{2\pi} [x_\beta(\xi) \cos \xi - y_\beta(\xi) \sin \xi] d\xi \\ A &= \frac{1}{2\pi} \int_0^{2\pi} [x_\infty(\xi) \cos \xi - y_\infty(\xi) \sin \xi] d\xi \end{aligned}\right\} \tag{34}$$

As is well known, the preceding scheme is an iterative numerical scheme. In lieu of this, we have developed a method which is fast and direct, and is equivalent to satisfying equation (32).

We circumscribe circles around the inner and outer boundary curves. Two cases arise depending on whether the circles are concentric or nonconcentric.

Case I: If the circumscribed circles are concentric (Fig. 2(a)), then we select those values of the ordinates which correspond to the abscissae

$$x_\beta = r_s \cos \xi, \; x_\infty = r_L \cos \xi \tag{35}$$

where r_s and r_L are the radii of the circumscribed circles in the physical plane.

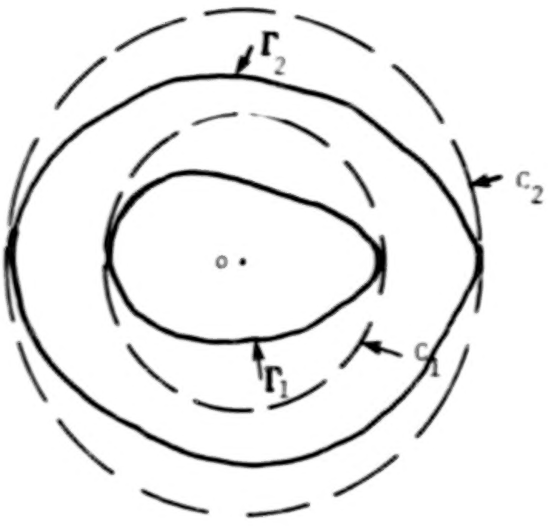
Case II: If the circumscribed circles are nonconcentric (Fig. 2(b)), then we first use the formula for the conformal transformation of nonconcentric to concentric circles, Kober [13], and choose the ordinates corresponding to abscissae given by the formula

$$\begin{aligned} x(\xi) = [&(1 - c\gamma \cos \xi)\{x_L(1 - c\gamma \cos \xi) + c\gamma y_L \sin \xi \\ &+ r_L(c \cos \psi - \gamma \cos(\xi-\psi))\} \\ &- c\gamma \sin \xi\{y_L(1 - c\gamma \cos \xi) - c\gamma x_L \sin \xi \\ &- r_L(c \sin \psi + \gamma \sin(\xi-\psi))\}] \\ &/(1 - 2c\gamma \cos \xi + c^2\gamma^2) \end{aligned} \tag{36}$$

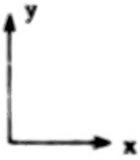
where

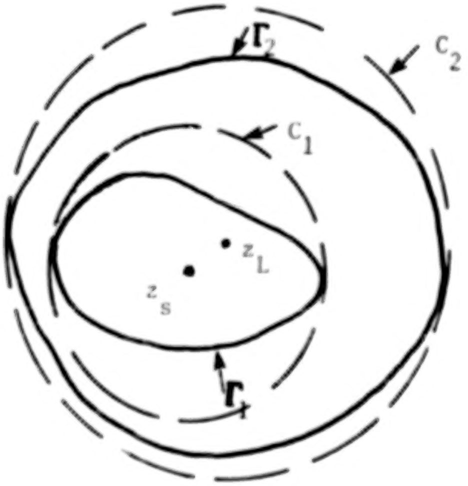
r_L, r_s = radii of outer and inner circumscribed circles

(x_L, y_L) and (x_s, y_s) = coordinates of the centers


Γ_2
C_2
o
C_1
Γ_1

(a) Concentric circumscribed circles C_1 and C_2 of radii r_s and r_L, respectively, with center at the origin.




Γ_2
C_2
C_1
z_L
z_s
Γ_1

(b) Nonconcentric circumscribed circles C_1 and C_2 of radii r_s and r_L and centers at z_s and z_L, respectively.

Figure 2.- Circumscribed circles.

$$d^2 = (x_s - x_L)^2 + (y_s - y_L)^2$$

$$\psi = \pi - \tan^{-1}\left(\frac{y_s - y_L}{x_s - x_L}\right)$$

$$c = [(d^2 + r_L^2 - r_s^2) + \{(d^2 + r_L^2 - r_s^2)^2 - 4d^2 r_L^2\}^{1/2}]/2r_L d$$

$\gamma = 1$ for the outer boundary

$\gamma = \frac{r_L}{r_s}\left|\frac{d-t}{t}\right|$ for the inner boundary

$$t = cr_L$$

Having determined the appropriate sets $(x_\beta(\xi),\ y_\beta(\xi))$ and $(x_\infty(\xi),\ y_\infty(\xi))$, we use (34) to obtain the values of a and A. The parametric difference η_∞ is connected in some manner with the "modulus" of the domain which, however, by itself is a separate problem (see Burbea [14] and Gaier [15]). In this work we have defined η_∞ based on the knowledge of a and A as discussed above by the formula

$$\eta_\infty = \ln\left(\frac{A}{a}\right) \tag{37}$$

For Figures 3 to 8, we have used the following functional forms of ϕ and f:

$$\phi(\bar{\xi}) = \frac{2\pi(\bar{\xi} - \bar{\xi}_o)}{\bar{\xi}_m - \bar{\xi}_o}$$

$$f(\bar{\eta}) = \frac{\eta_\infty(\bar{\eta} - \bar{\eta}_\beta)}{\bar{\eta}_\infty - \bar{\eta}_\beta} \quad \frac{K^{(\bar{\eta} - \bar{\eta}_\beta)}}{K^{(\bar{\eta}_\infty - \bar{\eta}_\beta)}}$$

so that

$$\lambda = \frac{2\pi}{\bar{\xi}_m - \bar{\xi}_o}$$

$$\theta = \frac{\eta_\infty}{\bar{\eta}_\infty - \bar{\eta}_\beta} [1 + (\bar{\eta} - \bar{\eta}_\beta) \ln K] \frac{K^{(\bar{\eta} - \bar{\eta}_\beta)}}{K^{(\bar{\eta}_\infty - \bar{\eta}_\beta)}}$$

where $K > 1$ is an arbitrary constant, and $\bar{\xi} = \bar{\xi}_m$, $\bar{\eta} = \bar{\eta}_\infty$ correspond, respectively, to $\xi = 2\pi$ and $\eta = \eta_\infty$. We treat $\bar{\xi}$ and $\bar{\eta}$ as integers so that $\bar{\xi}_o = 1$, $\bar{\xi}_m = \text{IMAX}$, $\bar{\eta}_\beta = 1$, and $\bar{\eta}_\infty = \text{JMAX}$. Since η_∞ is known from (37), hence by specifying the numerical values to K and JMAX we can create the desired mesh control in the direction of η. The value of K between 1.05 and 1.1 is quite sufficient [16] to have a fine grid near the inner boundary.

The number of terms to be retained in the series (24) is usually small for convex inner and outer boundaries, though we have retained (IMAX-1)/2 number of coefficients in each computation. This number is the optimum number of terms in a discrete Fourier series [17] having IMAX number of points in one period.

Figure 3 shows the classic case of confocal ellipses with coordinate contraction in η. The value of K is 1.05. The orthogonal correspondence between ξ-points of the inner and outer boundary has been established by using Case I, Eq. (35).

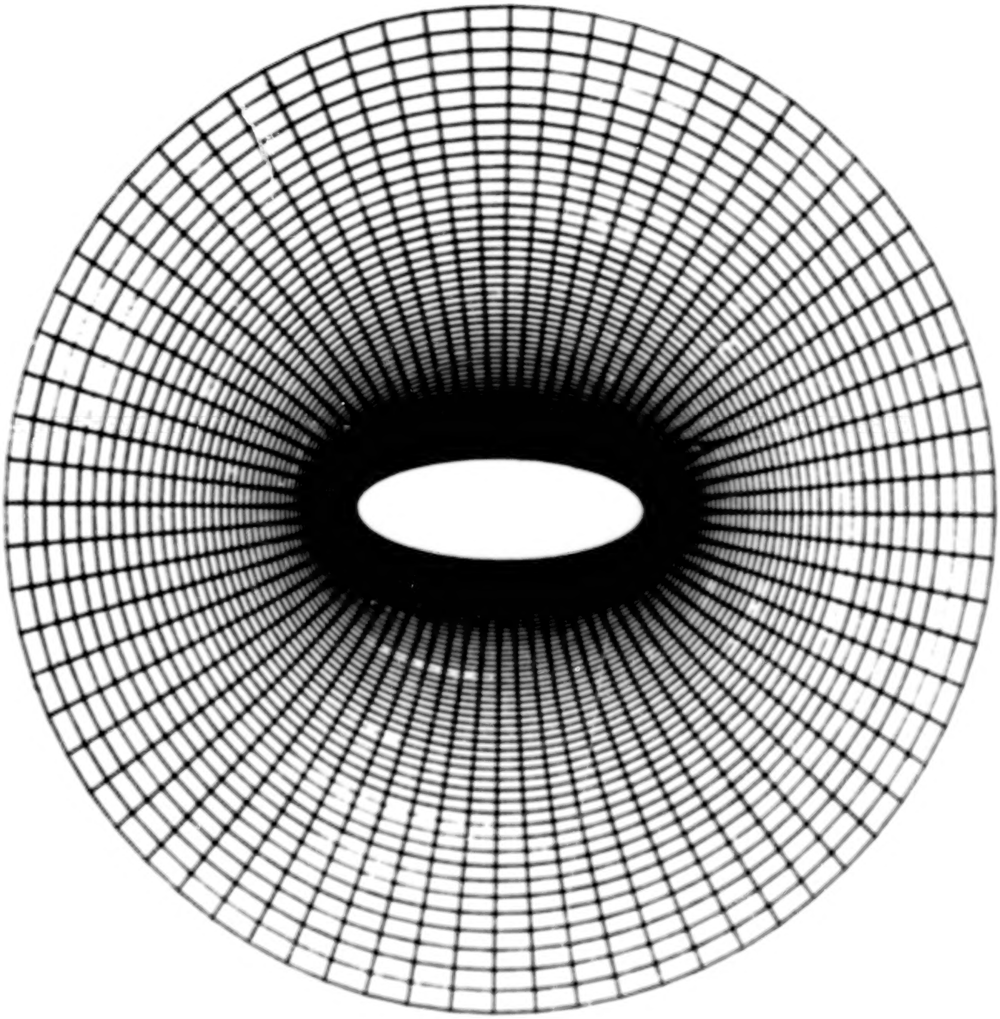


Figure 3.- Confocal ellipses. Semimajor axes 1.48, 5.0, and semiminor axes 0.5, 4.802, respectively. Only 38 η = Const. lines shown for detail.

Figure 4 presents orthogonal coordinates for a blunt body with elliptical outer boundary. Here $K = 1.01$. For orthogonal correspondence between ξ-points, Eq. (35) has been used.

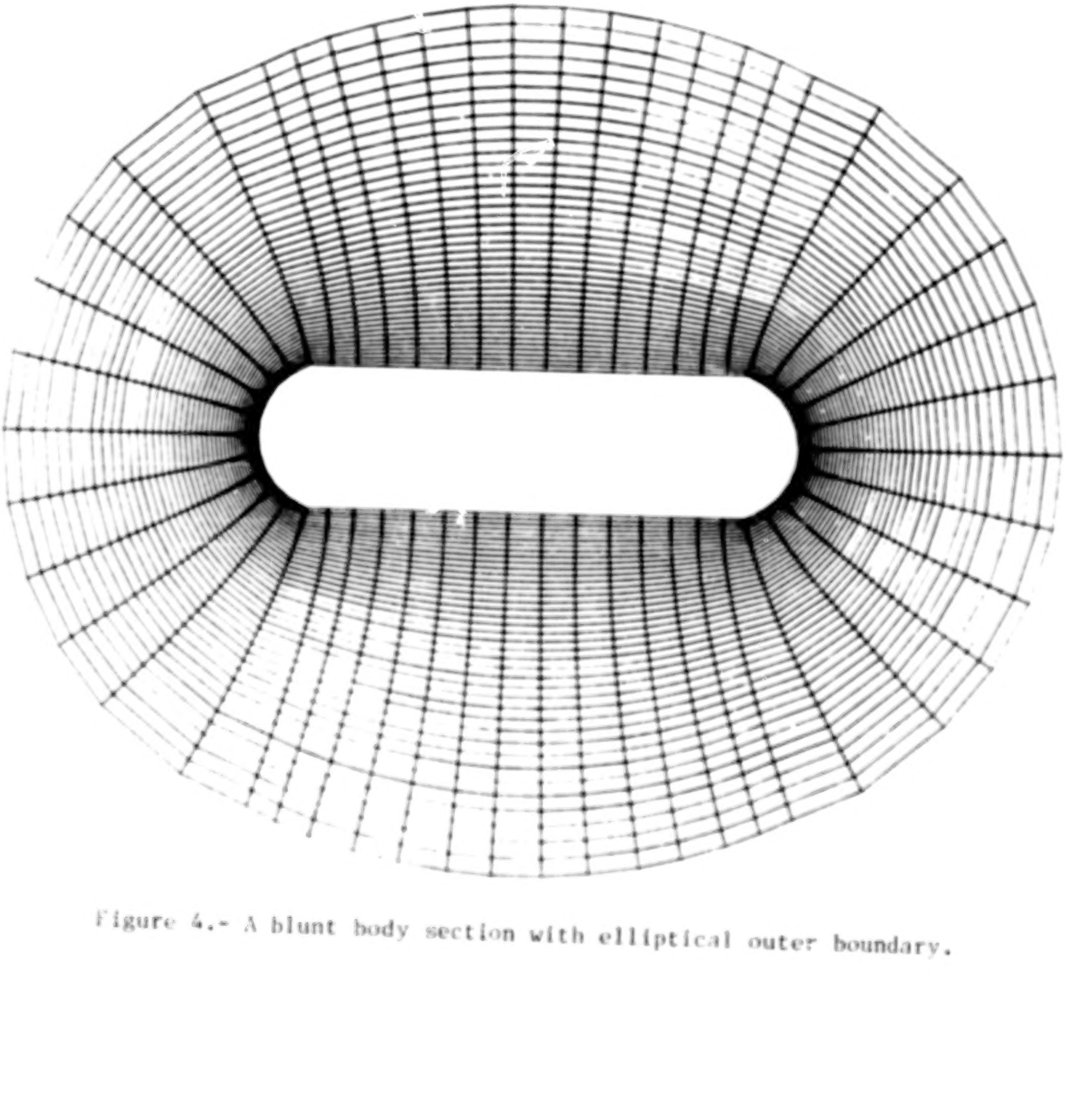


Figure 4.- A blunt body section with elliptical outer boundary.

Orthogonal coordinates for nonconcentric circles are presented in Figure 5. Here K = 1.01. For orthogonal correspondence between ξ-points between the inner and outer boundary, Eq. (36) has been used. Data shown on the figure.

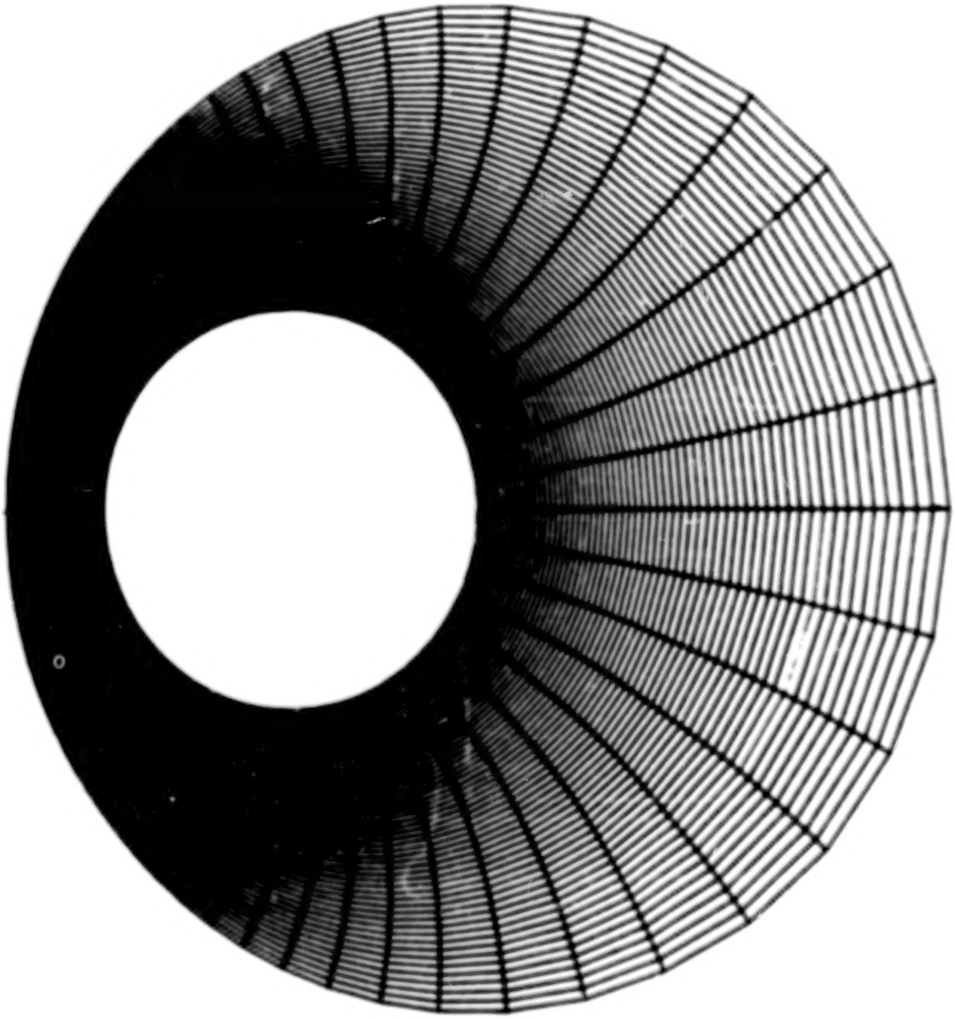


Figure 5.- Nonconcentric circles: $r_S = 1$, $r_L = 2.5$, $z_S = (0,0)$, $z_L = (1,0)$.

Orthogonal coordinates for a Joukowsky's airfoil with slightly rounded trailing edge are shown in Figure 6. Eq. (35) is used for orthogonal correspondence. Here K = 1.02.

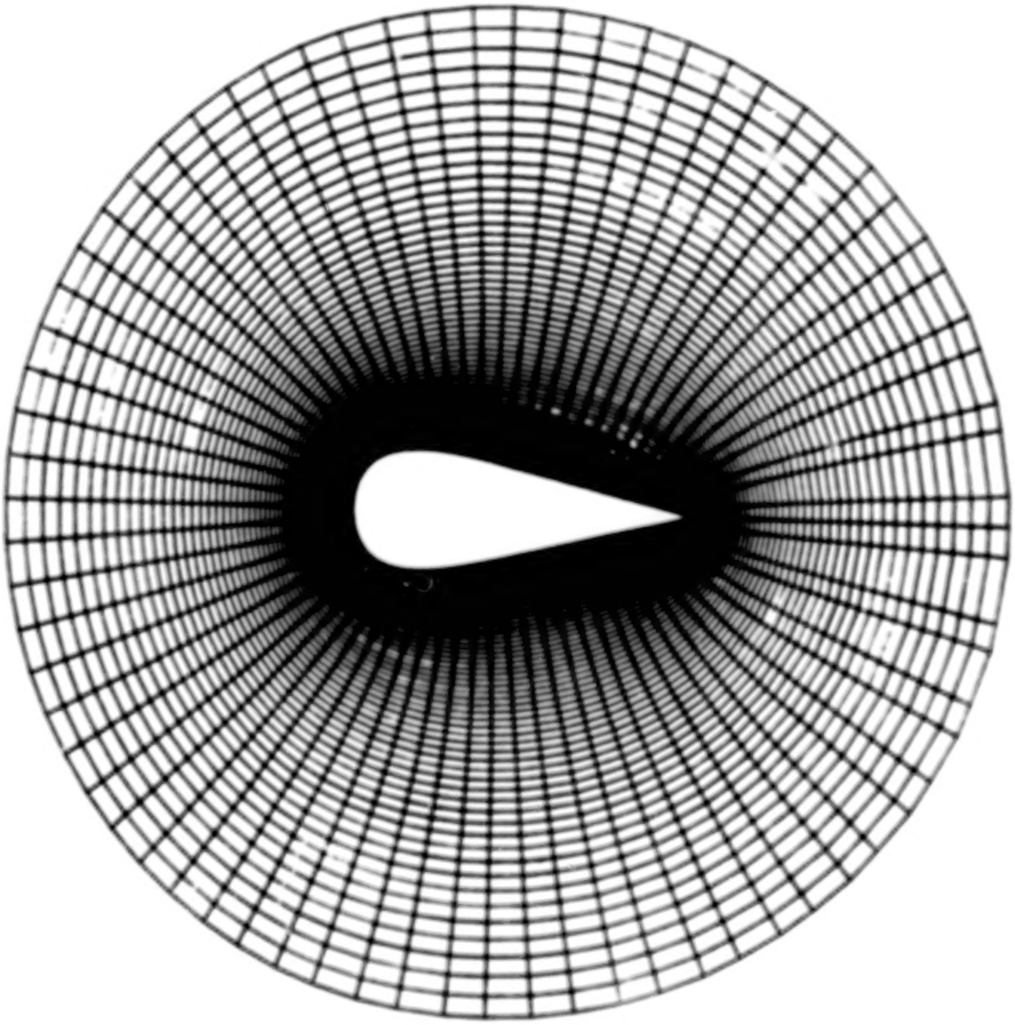


Figure 6.- Joukowsky's airfoil with slightly rounded trailing edge.

Figure 7 presents orthogonal coordinates for nonconcentric ellipses. Centers of the inner and outer ellipses are at (0,0) and (1,0), respectively. Here K = 1.01. For orthogonal correspondence Eq. (36) has been used.

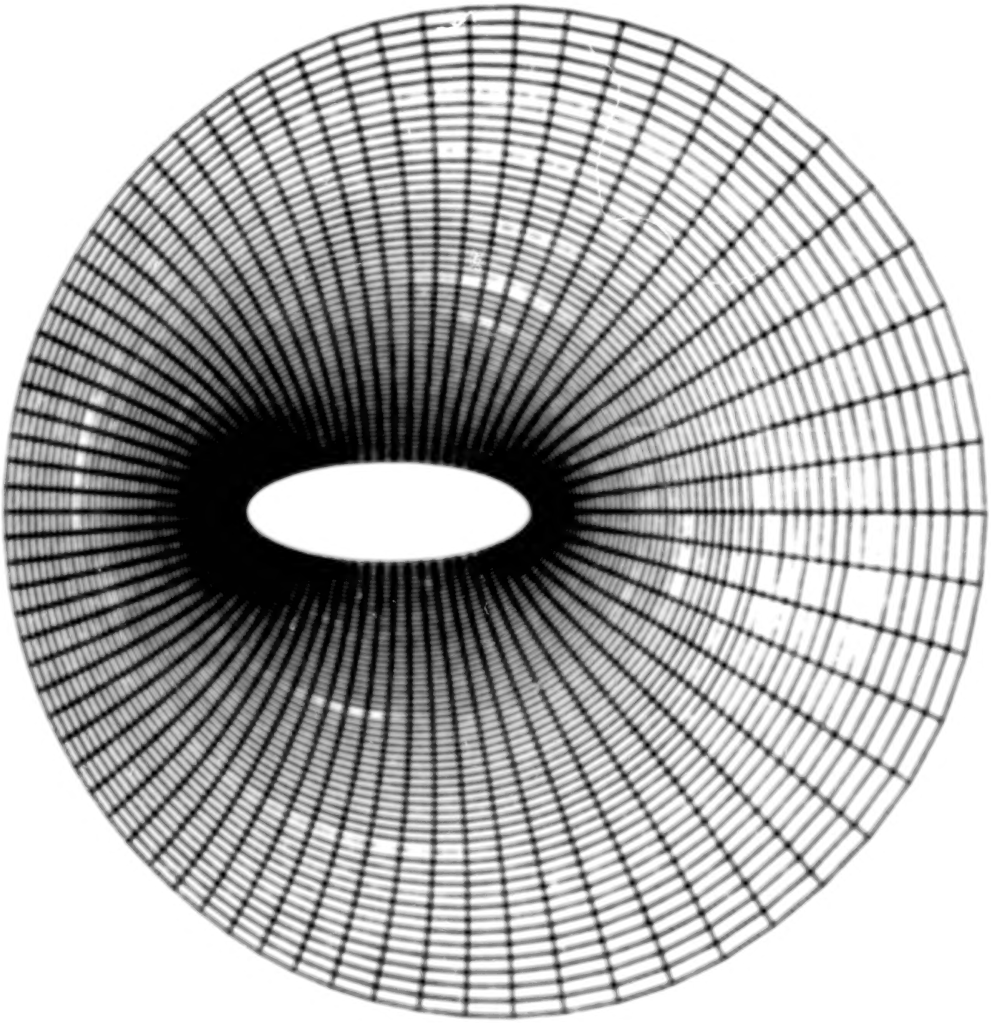


Figure 7.- Nonconcentric ellipses. Size data same as in Figure 3. $z_s = (0,0)$, $z_L = (1,0)$.

Orthogonal coordinates for an arbitrarily deformed upper part of Figure 4 are shown in Figure 8. The placement of outer boundary is limited to avoid intersecting normals (Eiseman [18]). This figure shows that we need some attraction near those sections of the outer boundary which face the concave side.

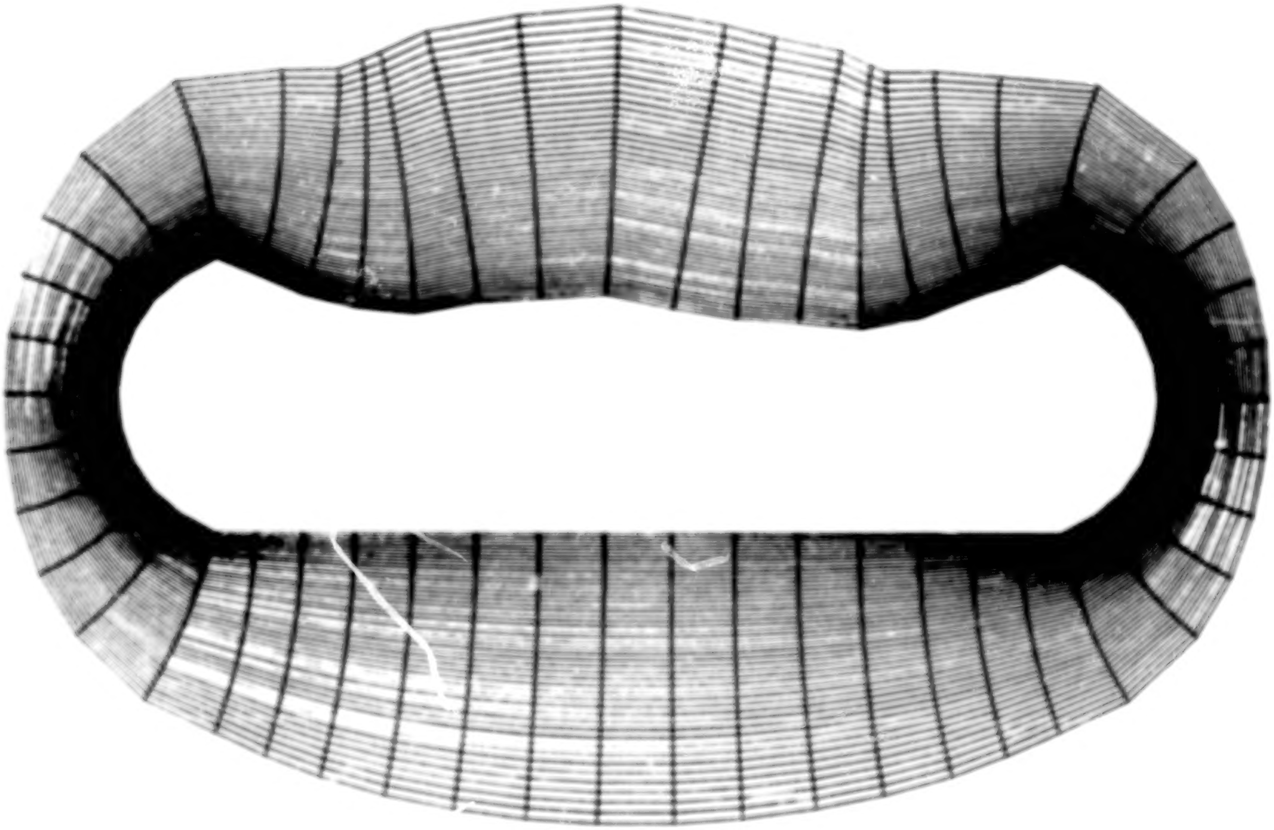


Figure 8.- Generated coordinates for body having convex, concave and straight portions. Placement of outer boundary is decided by the radius of the osculating circles of the concave portions.

Summary of Numerical Experimentation

In the course of this investigation a number of cases of inner and outer boundary shapes and orientations have been tested through the developed computer program. The main conclusions are listed below:

(i) The method works very effectively for smooth and convex boundaries of any shape and orientation.

(ii) For concave boundaries a method similar to that of Eiseman has to be used in the placement of the outer boundary to avoid intersecting normals. Another remedy would be to introduce some type of attraction near the outer boundary facing the concave side of the inner boundary.

(iii) Sharp turns and corners are not admissible and have to be rounded to avoid singularities in the metric data.

References

1. Z. U. A. Warsi, R. A. Weed, and J. F. Thompson, "Numerical Generation of Two-Dimensional Orthogonal Coordinate in an Euclidean Space", Engineering and Experimental Research Station, Mississippi State University, Rep. No. MSSU-EIRS-ASE-80-3, June 1980.

2. A. J. Winslow, "Numerical Solution of the Quasi-Linear Equation in a Non-Uniform Triangular Mesh", Journal of Computational Physics, 2, 149 (1966).

3. W. D. Barfield, "An Optimal Mesh Generator for Lagrangian Hydrodynamic Calculations in Two Space Dimensions", Journal of Computational Physics, 6, 417 (1970).

4. W. H. Chu, "Development of a General Finite Difference Approximation for a General Domain, Part I: Machine Transformation", Journal of Computational Physics, 8, 392 (1971).

5. A. A. Amsden and C. W. Hirt, "A Simple Scheme for Generating General Curvilinear Grids", Journal of Computational Physics, 11, 348 (1973).

6. D. E. Potter and G. H. Tuttle, "The Construction of Discrete Orthogonal Coordinates", Journal of Computational Physics, 13, 483 (1973).

7. J. F. Thompson, F. C. Thames, and C. W. Mastin, "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies", Journal of Computational Physics, 15, 299 (1974).

8. S. B. Pope, "The Calculation of Turbulent Recirculating Flows in General Orthogonal Coordinates", Journal of Computational Physics, 26, 197 (1978).

9. J. F. Middlecoff and P. D. Thomas, "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", AIAA Computational Fluid Dynamics Conference, Paper No. 79-1462 (1979).

10. R. L. Sorenson and J. L. Steger, "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations", NASA-TM-73252, August 1977.

11. G. Starius, "Constructing Orthogonal Curvilinear Meshes by Solving Initial-Value Problem", Numerische Mathematik, 28, 25 (1977).

12. A. Cohen, "An Introduction to the Lie Theory of One-Parameter Groups", G. E. Stechert and Co., New York (1931), p. 2-5.

13. H. Kober, "Dictionary of Conformal Representations", Dover Publications, Inc. (1952).

14. J. Burbea, "A Numerical Determination of the Modulus of Doubly Connected Domains by Using the Bergman Curvature", Mathematics of Computation, 25, 743 (1971).

15. D. Gaier, "Determination of Conformal Modulus of Ring Domains and Quadrilaterals", In-Lecture Notes in Mathematics, No. 399, Springer-Verlag, Berlin (1974).

16. Z. U. A. Warsi and J. F. Thompson, "Machine Solutions of Partial Differential Equations in the Numerically Generated Coordinate Systems", Engineering and Experimental Research Stateion, Mississippi State University, Rep. No. MSSU-EIRS-ASE-77-1, August (1976).

17. F. B. Hildebrand, "Introduction to Numerical Analysis", McGraw-Hill Book Company, Inc., (1956).

18. P. R. Eiseman, "A Coordinate System for a Viscous Transonic Cascade Analysis", Journal of Computational Physics, 26, 307 (1978).

BLANK PAGE

BLANK PAGE

A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-SPACING CONTROL

J. C. Wu
and
U. Gulcat

Georgia Institue of Technology

It is demonstrated by analyses and by numerical illustrations that any arbitrarily prescribed contour, open or closed, can be mapped conformally onto a simple contour, such as a unit circle, using any arbitrarily prescribed distribution of scale factor of transformation. This flexibility of selecting a scale factor distribution on the contour is not in violation of the well-known Riemann's uniqueness theory for conformal mapping. The much used Joukowski transformation is shown to be one of a family of conformal transformations that map a given airfoil contour onto a unit circle. For flow problems, the conformal mapping of a region bounded by a complicated contour onto a corresponding region bounded by a simple contour is of interest. With an arbitrarily prescribed scale factor, there exist in general singular points located at finite distances from the contour. (The case where singularities are located infinitely far from the contour is an exception.) Numerical methods for generating conformal grids should therefore incorporate a mechanism that ensures the absence of singular points in the region of interest. In this context, the distribution of scale factor on the contour cannot be arbitrary. The restriction on the scale factor distribution is not stringent. There exists ample freedom in the control of grid spacing on the contour so that, in general, the physics of the flow problem can be accommodated by a suitably designed conformal grid.

DESIRED FEATURES OF GRID SYSTEMS

- Be body-fitted.
- Possess control over grid-spacing.
- Yield algebraic equations amenable to highly efficient numerical procedures.
- Require minimal computational efforts to generate.

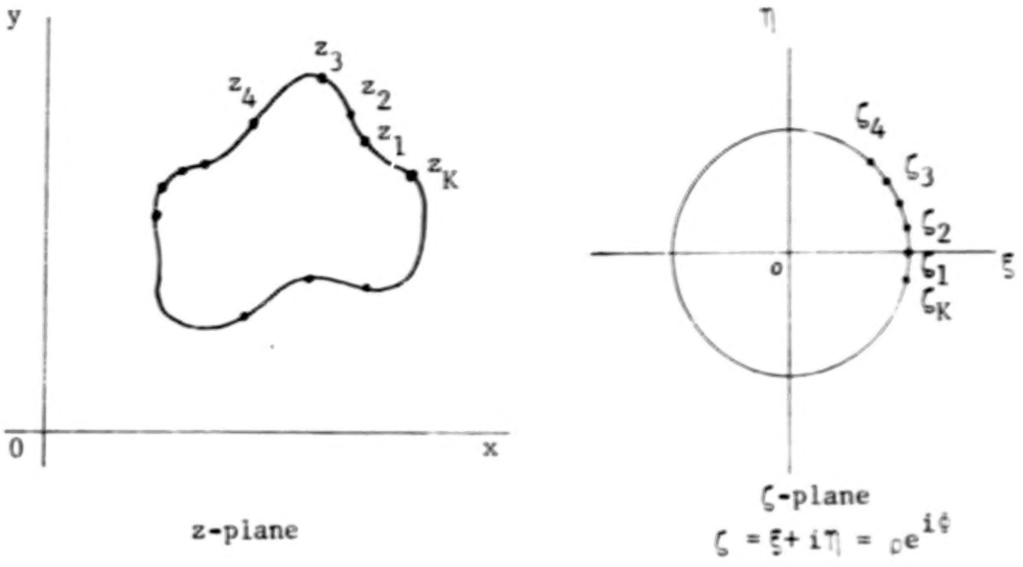
The first feature listed above is generally accepted as being the key to the successful computation of flows. The second feature is essential to the computation of complex flows with diverse length scales in different regions of the flows. The third feature is critical in situations where the amount of computation required is very large. The fourth feature is important if repeated generation of grids is desired during the solution of a given problem. (For example, in the solution of a time-dependent problem, different grids may be desired for different time intervals).

CURVILINEAR COORDINATES AND GRID SYSTEM

- Non-orthogonal coordinates yield transformed differential equations that are substantially more complicated than the original equations.
- Orthogonal non-conformal coordinates yield less complicated equations.
- Conformal coordinates yield simplest transformed equations.
- The requirements that a transformation be conformal and that it possesses a grid-spacing-control ability are not mutually exclusive.
- Conformal mapping can be generated very efficiently.
- Orthogonal grids can be easily developed using conformal mapping.

The advantages of using conformal grids are most clearly demonstrated by the numerical procedures available for the Poisson's equation. Algebraic equations obtained in conformal grids can be solved using direct methods such as the block Gaussian elimination, the odd-even reduction, and the Fourier series methods. The choice of methods is somewhat more limited in an orthogonal non-conformal grid. With non-orthogonal grids, iterative procedures are generally required. The main purpose of this paper is to show that any prescribed two-dimensional body contour can be conformally mapped onto a simple shape, such as the unit circle, and such mappings do possess a grid-spacing-control ability.

CONFORMAL MAPPING


y
0
x
z_1
z_2
z_3
z_4
z_K
z-plane
η
ξ
o
ζ_1
ζ_2
ζ_3
ζ_4
ζ_K
ζ-plane
$\zeta = \xi + i\eta = \rho e^{i\phi}$

Any usual contour, open or closed, can be mapped conformally onto a simple contour, such as a circle or a straight line segment, using any prescribed distribution of the scale factor of transformation on the contour. This is true for smooth contours as well as for contours with discontinuous slopes. The unit circle is used as the canonical contour for the following discussion. A total of K equally spaced points are assigned on the unit circle, with the point ζ_k given by

$$\zeta_k = e^{i2\pi k/K}$$

where K is an odd integer.

The corresponding points, z_k, on the original contour are sequenced as shown but otherwise arbitrarily located.

THE LAURENT SERIES

$$\zeta = \rho \, e^{i\varphi}$$

$$\zeta_k = \exp\left(\frac{i2\pi k}{K}\right) \quad , \quad K \text{ an odd integer}$$

$$z = f(\zeta) = \sum_{n=-N}^{N} C_n \zeta^n \quad , \quad N = \frac{K-1}{2}$$

$$z_k = \sum_{n=-N}^{N} C_n \exp\left\{\frac{i2\pi kn}{K}\right\}$$

$$\sum_{k=-N}^{N} z_k \exp\left\{\frac{i2\pi km}{K}\right\} = \sum_{n=-N}^{N} C_n \sum_{n=-N}^{N} \exp\left\{\frac{i2\pi k(m-n)}{K}\right\}$$

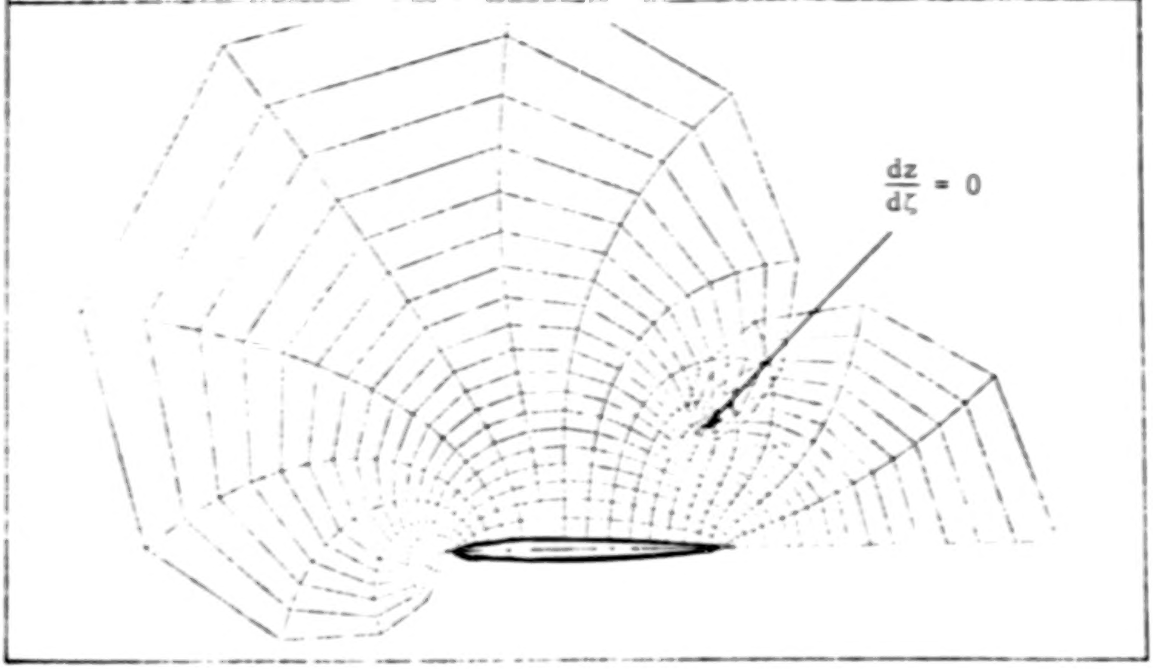
$$= \begin{cases} K & m=n \\ 0 & m=n \end{cases}$$

$$C_n = \frac{1}{K} \sum_{k=-N}^{N} z_k \exp\left\{\frac{i2\pi kn}{K}\right\}$$

$$z = \frac{1}{K} \sum_{n=-N}^{N} \sum_{k=-N}^{N} z_k \exp\left\{\frac{i2\pi kn}{K}\right\} \zeta^n$$

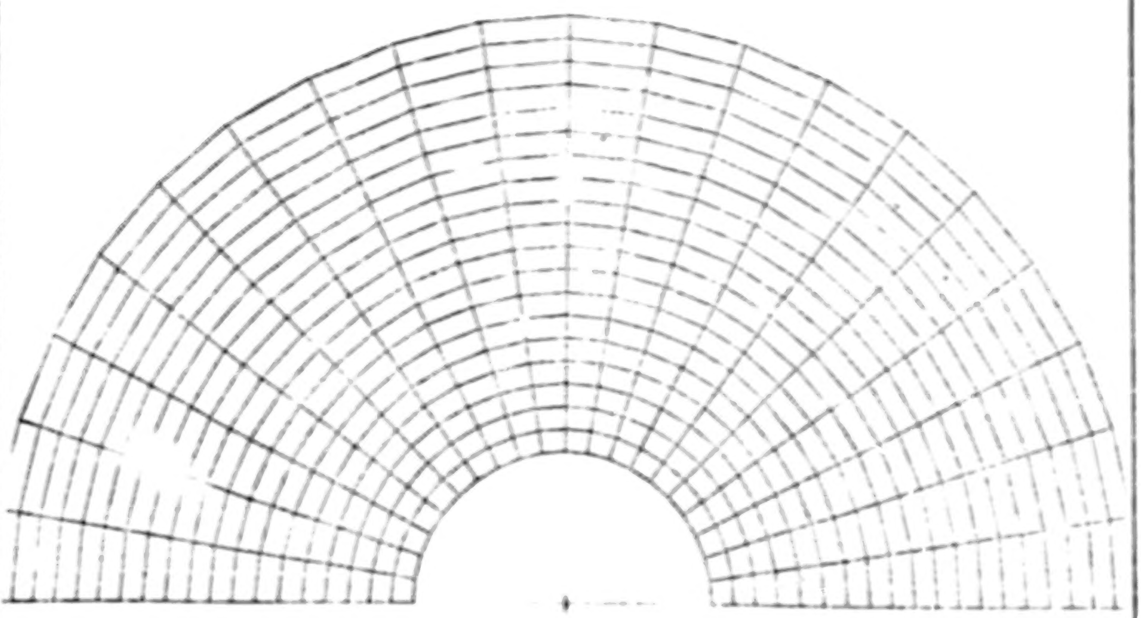
By analytic continuation, the Fourier coefficients of the Laurent series are those obtained above. The finite Laurent series therefore can be used to compute the grid-point locations away from the contour that corresponds to specified grid points in the ζ-plane. The above analysis can be carried out for an infinite Laurent series. The only change is that the Fourier coefficients are then expressed as integrals instead of sums. The finite Laurent series represents an approximation of the infinite Laurent series whose regular part converges inside a certain circle and whose principal part converges outside another certain circle. The domain of convergence of the infinite Laurent series is the common annulus of the two circles. The finite Laurent series produces accurate conformal grids in this domain of convergence. The conformality of the grids thus generated is ensured by the analyticity of the Laurent series.

A CONFORMAL GRID WITH SINGULAR POINTS
NEAR THE CONTOUR



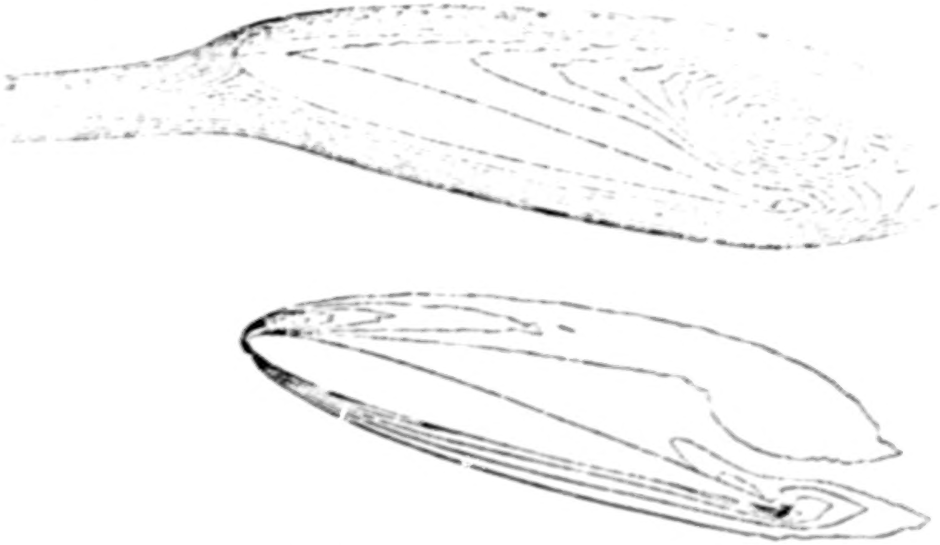
With an arbitrarily prescribed distribution of the scale factor, there exist in general "singular points" located at finite-distances from the contour. Therefore, numerical methods for generating conformal grids should contain provisions that ensure the absence of singular points in the region of computational interest. In this context, the distribution of the scale factor on the contour cannot be arbitrary. In this figure is shown a grid around a symmetric airfoil with singular points located near the airfoil. This figure is obtained using the finite Laurent series method. The prescribed points on the airfoil are symmetrically distributed about the line of symmetry of the airfoil. The grid lines shown are mapped onto the radial lines and concentric circles shown on the next figure.

THE GRID SYSTEM IN THE CIRCLE-PLANE



The "canonical" domain used here is the domain exterior to the unit circle. The grid lines shown here are mapped conformally onto the grid lines shown in the airfoil-planes at all points except the singular points where the mapping ceases to be conformal. The gird shown is orthogonal with equal spacings in the angular and the radial directions.

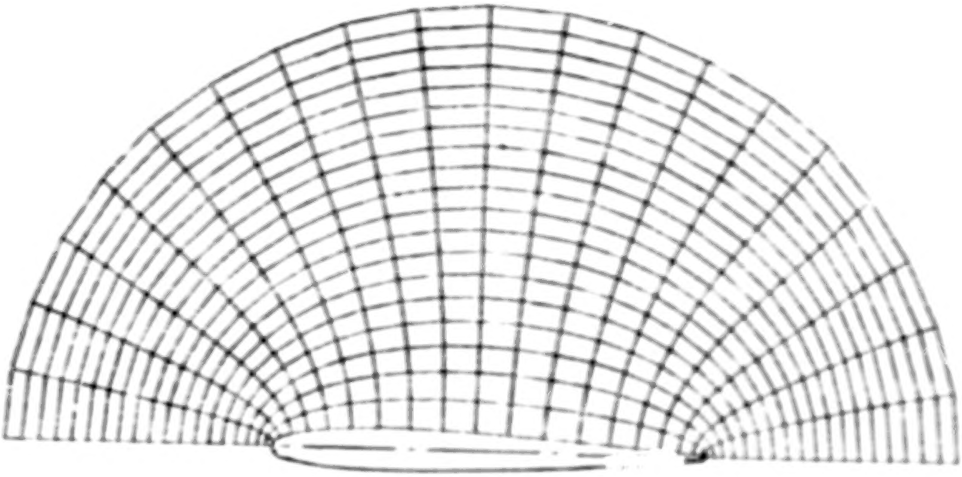
SOLUTION FIELD AROUND AN AIRFOIL



12% Thick Joukowski Airfoil
Re = 3.63 x 10^6, $\alpha = 15^\circ$, t = 13.2

Here are shown some streamlines and constant vorticity lines around a 9% thick symmetric airfoil. With the integro-differential approach the authors are using, the solution field can be confined to the vortical region of the flow. In consequence, the presence of the singular points in the conformal grid outside the vortical region is acceptable. With prevailing method, the flowfield is usually truncated and it is permissible to have singular points present outside the truncated region.

JOUKOWSKI GRID

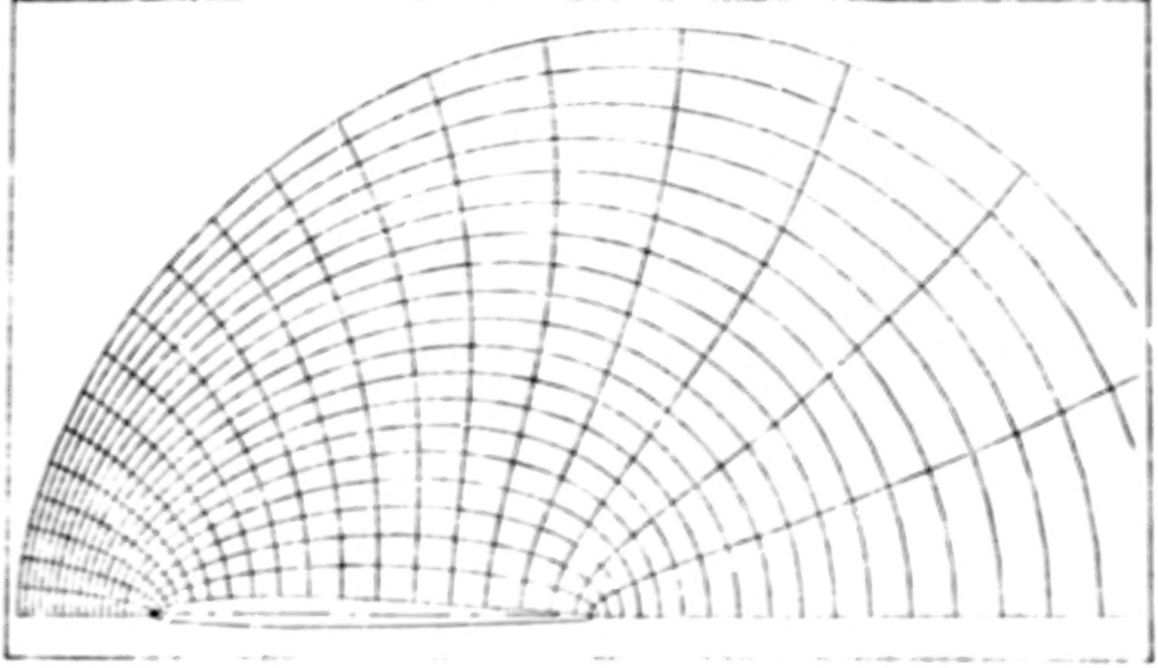


This figure shows the grid lines around a 9 % thick symmetric airfoil that is mapped using the Joukowski transformation

$$z = \zeta - 0.05214 + \frac{0.854078}{\zeta - 0.05214}$$

With this transformation there is no singular point at a finite distance from the airfoil. The trailing edge in this transformation is rounded (so as to avoid the need of the Schwarz-Christoffel procedure, which would have introduced complications unnecessary at this stage of development). The finite Laurent series method, with grid points on the airfoil boundary assigned properly, produces a grid system that is indistinguishable from the one shown.

A SYMMETRIC BILINEAR GRID

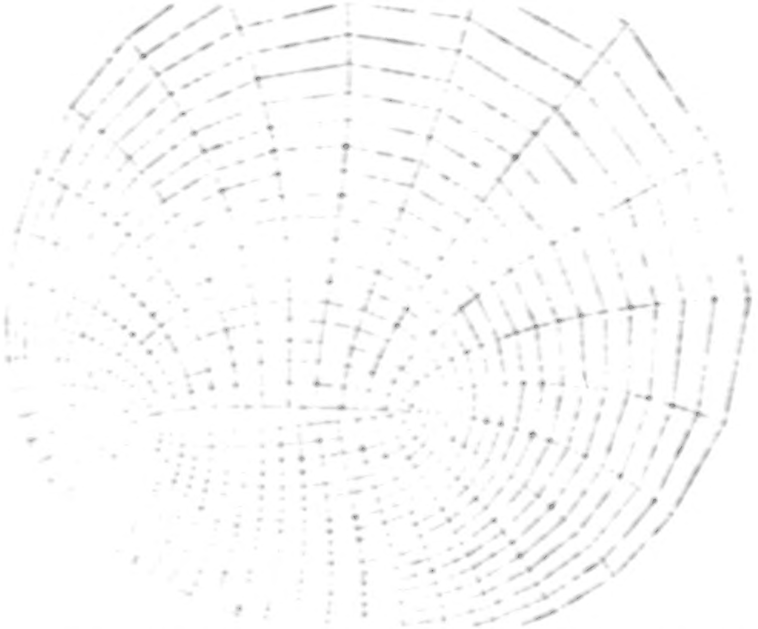


The unit circle in the ζ -plane can be mapped onto a unit circle in the w-plane through a bilinear transformation of the form

$$\zeta = \frac{w - \alpha}{1 - \alpha w}$$

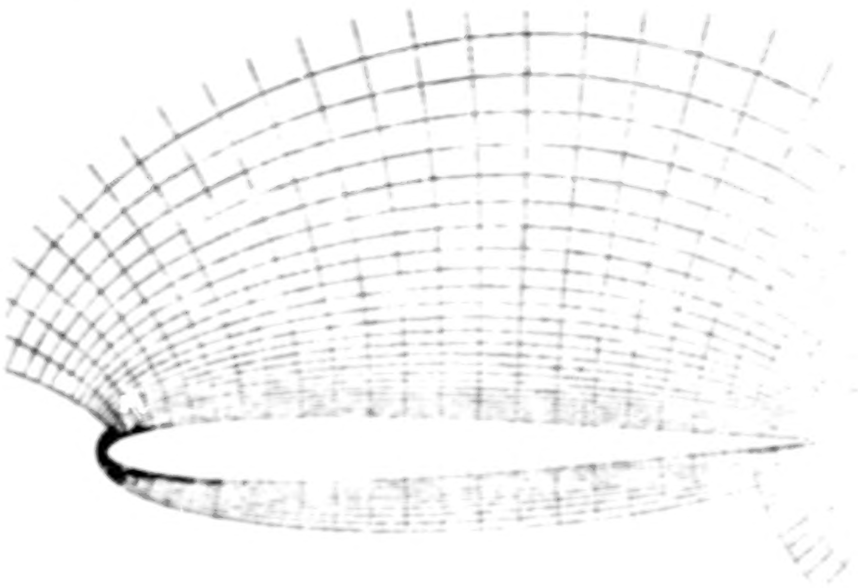
For any assigned value of α , the points on the airfoil boundary that correspond to uniformly distributed grid points on the unit circle can be located. The finite Laurent series method then yields a conformal mapping of a region exterior of the circle in the w-plane onto a region exterior of the airfoil in the z-plane. The concentric circles and radial lines in the w-plane are mapped onto the grid lines shown above for the case $\alpha = 1/8$. The grid lines are symmetric about the line of symmetry of the airfoil.

A NON-SYMMETRIC BILINEAR GRID



Using a complex value for α, the concentric circles and radial lines in the w-plane are mapped onto non-symmetric grid lines in the airfoil-plane. The figure above shows grid lines for the case $\alpha=\frac{1+i}{\sqrt[4]{2}}$. The singular points of the grid system shown here and in the previous figure are sufficiently far from the airfoil so that the grids are of practical interest.

A GRID IN USE



This grid has been used in a computation of a flow past a 9 % thick symmetric airfoil at an angle of attack of 15%. The grid is a bilinear grid with $\sigma = 16$. In this study, the boundary layer region of the flow is computed separately from the detached region. It is only necessary to generate a grid covering the computation field and to keep the singular point away from this computation field.

CONTENTS

PREFACE . . . . iii 1/A7

1. GRID GENERATION USING CLASSICAL TECHNIQUES 1 1/B1
Gino Moretti

2. GRID GENERATION USING DIFFERENTIAL SYSTEMS TECHNIQUES 37 1/D6
Joe F. Thompson and C. Wayne Mastin

3. MESH GENERATION USING ALGEBRAIC TECHNIQUES 73 1/G3
Peter R. Eiseman and Robert E. Smith

4. GENERATION OF BOUNDARY AND BOUNDARY-LAYER FITTING GRIDS 121 2/C1
C. M. Ablow and S. Schechter

5. AN ELECTROSTATIC ANALOG FOR GENERATING CASCADE GRIDS 129 2/D6
John J. Adamczyk

6. FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION 143 2/E6
Peter G. Anderson and Lawrence W. Spradley

7. COMPONENT-ADAPTIVE GRID EMBEDDING 157 2/F6
E. H. Atta

8. GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS 175 2/G10
A. J. Baker and P. D. Manhardt

9. GRID GENERATION FOR TIME DEPENDENT PROBLEMS: CRITERIA AND METHODS 181 3/A5
Marsha Berger, William Gropp, and Joseph Oliger

10. GENERATIONS OF ORTHOGONAL SURFACE COORDINATES 189 3/A1
F. G. Blottner and J. B. Moreno

11. AN ADAPTIVE COMPUTATION MESH FOR THE SOLUTION OF SINGULAR PERTURBATION PROBLEMS 193 3/B3
J. U. Brackbill and J. Saltzman

12. A NEW COORDINATE TRANSFORMATION FOR TURBULENT BOUNDARY LAYER FLOWS 197 3/B7
J. E. Carter, D. E. Edwards, and M. J. Werle

13. GENERATION OF ORTHOGONAL BOUNDARY-FITTED COORDINATE SYSTEMS 213 3/C9
Roderick M. Coleman

14. NONLINEAR GRID ERROR EFFECTS ON NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS 221 3/D3
S. K. Dey

Page

15. GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS 231 3/D1
Lawrence J. Dickson

16. FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL
TURBOMACHINERY FLOW CALCULATIONS 241 3/E9
Djordje S. Dulikravich

17. THREE-DIMENSIONAL SPLINE-GENERATED COORDINATE TRANSFORMATIONS
FOR GRIDS AROUND WING-BODY CONFIGURATIONS 253 3/F7
Lars-Erik Eriksson

18. AN INVESTIGATION INTO GRID PATCHING TECHNIQUES 265 3/G5
C. R. Forsey, M. G. Edwards, and M. P. Carr

19. BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED
BOUNDARIES AND REENTRANT BOUNDARIES 295 4/B1
U. Ghia and K. N. Ghia

20. A SIMPLE NUMERICAL ORTHOGONAL COORDINATE GENERATOR FOR
FLUID DYNAMIC APPLICATIONS 307 4/C8
Randolph A. Graves, Jr.

21. A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR
FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS 315 4/D2
Darryl W. Hall

22. CONFORMAL MAPPINGS OF MULTIPLY CONNECTED REGIONS ONTO
REGIONS WITH SPECIFIED BOUNDARY SHAPES 329 4/G2
Andrew N. Harrington

23. BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY
COMPUTATIONAL REGIONS 331 4/E4
Edward J. Kowalski

24. GRID GENERATION FOR GENERAL THREE-DIMENSIONAL CONFIGURATIONS 355 4/F1
K. D. Lee, M. Huang, N. J. Yu, and P. E. Rubbert

25. EFFECT OF GRID SYSTEM ON FINITE ELEMENT CALCULATION 367 4/G1
K. D. Lee and S. M. Yen

26. SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX
FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN
SHOCK LAYER AND BASE FLOW 377 5/A
C. K. Lombard, M. P. Lombard, G. P. Menees, and J. Y. Yang

27. AN ANALYTICAL TRANSFORMATION TECHNIQUE FOR GENERATING
UNIFORMLY SPACED COMPUTATIONAL MESH 385 5/B
Yoon H. Oh

Page

28. APPLICATION OF THE MULTIGRID METHOD TO GRID GENERATION 399 5/C5
Samuel Ohring

29. GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS 409 5/D1
Man Mohan Rai and D. A. Anderson

30. A TWO DIMENSIONAL MESH VERIFICATION ALGORITHM 431 5/E9
R. Bruce Simpson

31. GENERATION OF C-TYPE CASCADE GRIDS FOR VISCOUS
FLOW COMPUTATION 437 5/F1
Peter M. Sockol

32. NUMERICAL GENERATION OF TWO-DIMENSIONAL GRIDS BY THE USE
OF POISSON EQUATIONS WITH GRID CONTROL AT BOUNDARIES 449 5/F13
Reese L. Sorenson and Joseph L. Steger

33. USE OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS TO
GENERATE BODY FITTED COORDINATES 463 5/G13
Joseph L. Steger and Reese L. Sorenson

34. CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS 479 6/B4
Frank B. Tatom, William R. Waldrop, and S. Ray Smith

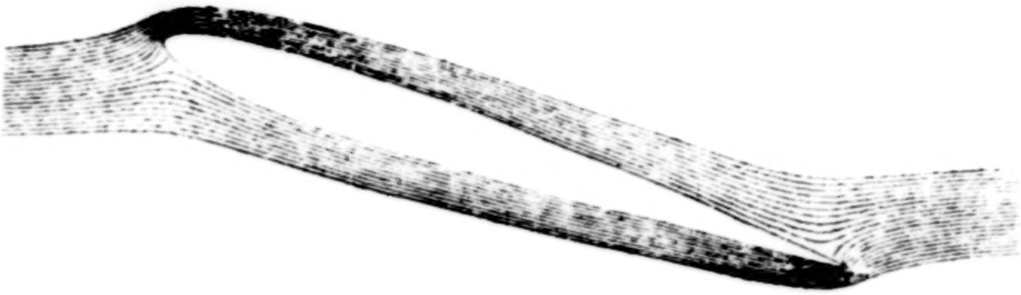
35. GRID GENERATION FOR TWO-DIMENSIONAL FINITE ELEMENT
FLOWFIELD COMPUTATION 505 5/D5
Kenneth E. Tatum

36. NUMERICAL GENERATION OF TWO-DIMENSIONAL ORTHOGONAL
CURVILINEAR COORDINATES IN AN EUCLIDEAN SPACE 519 5/E5
Z. U. A. Warsi and J. F. Thompson

37. A BODY-FITTED CONFORMAL MAPPING METHOD WITH GRID-
SPACING CONTROL 545 5/G3
J. C. Wu and U. Gulcat

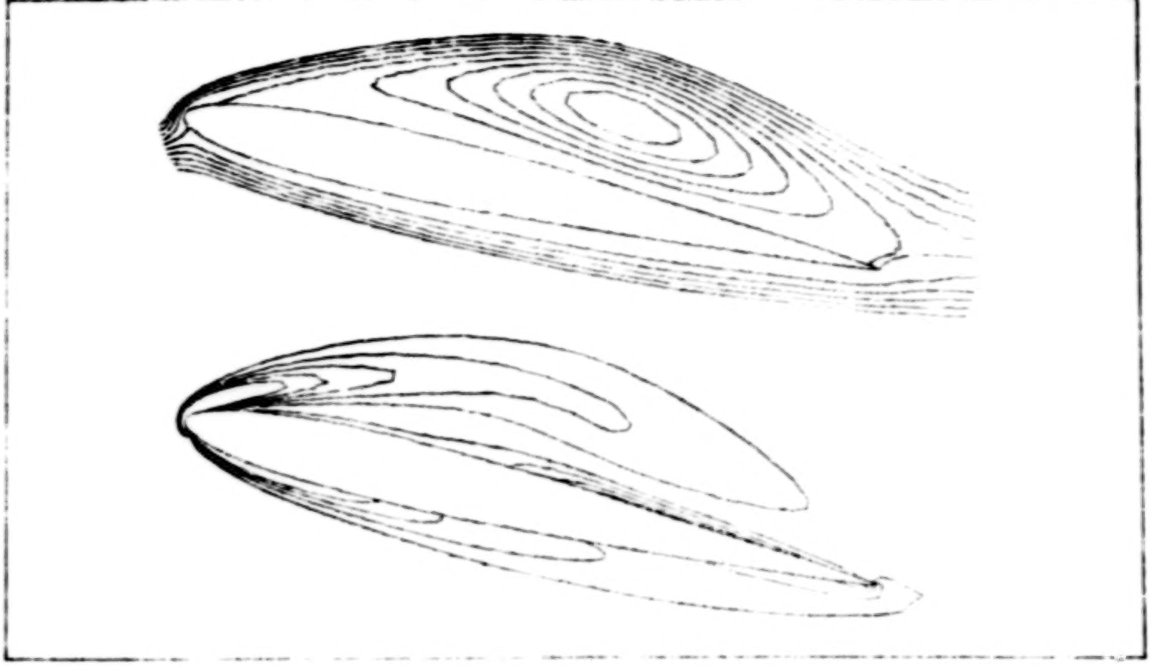
ATTENDEES 561 7/A3

INITIAL FLOW PATTERN AROUND AN AIRFOIL



This and the following figures show computed streamlines and vorticity contours around a 9 % thick airfoil set into motion impulsively and thereafter kept moving at a constant velocity with an angle of attack of 15° and a Reynolds number of 1000. This figure is for the time level immediately after the motion's onset. The vorticity is confined to the boundary of the airfoil and the flow away from the airfoil is potential. Note that the rear stagnation point is on the upper surface of the airfoil.

FLOW AROUND AN AIRFOIL WITH A SEPARATION BUBBLE



This figure shows the computed streamlines and constant vorticity contours around the airfoil after the airfoil has advanced 2.9 chord lengths relative to the freestream. A separation bubble has appeared and grown to its present size. The vorticity field is still confined to the region near the airfoil as shown. With the integro-differential approach used here, it is only necessary to perform computations in the vortical region. Therefore the grid needs only be generated for the vortical region.

OUTLINE OF GRID GENERATION PROCEDURE

The recommended procedure for generating a conformal grid with control over grid spacing consists of four steps:

(1) The locations of grid points on the physical contour that are mapped onto equally spaced points on a unit circle through a "Joukowski type" conformal transformation are computed.

(2) The coefficients in a finite Laurent series are computed as described earlier.

(3) A suitable bilinear transformation is introduced.

(4) Grid locations corresponding to concentric circles and radial lines in the bilinear transformed plane are computed.

A computer program (prepared by N. L. Sankar) which performs step (1) is available. This program uses an iterative procedure (Bauer et al, 1977 and other researchers). A spline approximation is utilized to achieve a high degree of accuracy. The operation count for this step is small. For each given contour, if several different grids are to be generated, then step (4) is the only step that needs to be repeated. Steps (1) and (2) need to be performed only once for the contour. Step (3) needs to be performed only once for all contours of interest.

CONCLUDING REMARKS

- Body-fitted conformal grids can be generated efficiently using the approach described.
- Ample freedom exists in the control of grid spacing on any contour so that the physics of the flow can be suitably accomadated.

The work reviewed here represents only the initial stage of development of a new conformal mapping approach for grid generation. Based on the results obtained thus far, this approach is a highly promising one for use in computing complex flow problems.

ATTENDEES

Michael Abbett
Acurex Corporation

C. M. Ablow
SRI International

John J. Adamczyk
NASA Lewis Research Center

D. A. Anderson
Department of Aerospace Engineering
Iowa State University

John D. Anderson, Jr.
Department of Aerospace Engineering
University of Maryland

Joshua Anyiwo
NASA Langley Research Center

E. Atta
Lockheed Georgia Company

A. J. Baker
University of Tennessee

R. Balasubramanian
NASA Langley Research Center

Norman Banks
U.S. Army Ballistic Research Lab

Bob Bennett
NASA Langley Research Center

Marsha Berger
Department of Computer Science
Stanford University

Thomas J. Barber
Pratt & Whitney Aircraft

Sam Bland
NASA Langley Research Center

F. G. Blottner
Sandia National Lab

Percy J. Bobbitt
NASA Langley Research Center

Jeremiah U. Brackbill
Los Alamos Scientific Laboratories

John Carlson
NASA Langley Research Center

M. P. Carr
Aircraft Research Association Ltd.

J. E. Carter
Computational Fluid Dynamics Group
United Technologies Research Center

Sukumar Chakravarthy
Rockwell International Science Center

I-Shih Chang
The Aerospace Corporation

L. T. Chen
McDonald Douglas Research Laboratories

Don Chenoweth
Sandia Laboratory

Roderick M. Coleman
David W. Taylor Naval Ship Research
and Development Center

Bill Compton
NASA Langley Research Center

Raul J. Conti
Lockheed Research Lab

Chris Cox
NASA Langley Research Center

Charlotte Craidon
NASA Langley Research Center

Herb Cunningham
NASA Langley Research Center

John Dannenhoffer
Pratt & Whitney Aircraft

Fred Dejarnette
Mechanical & Aerospace
Engineering Department
N. C. State University

Robert A. Delaney
Detroit Diesel Allison
Division General Motors Corp.

Bob Desmarais
NASA Langley Research Center

Krishna Deva-Rayalu
Boeing Commercial Airplane Division

S. K. Dey
Mathematics Department
Eastern Illinois University

Larry Dickson
Department of Aeronautics & Astronautics
University of Washington

Bill Dietz
Arnold Research Organization

Mich Doria
NASA Langley Research Center

Phil Drummond
NASA Langley Research Center

Douglas Dwoyer
NASA Langley Research Center

D. E. Edwards
Computational Fluid Dynamics Group
United Technologies Research Center

James Edwards
Knolls Atomic Power Lab

John Edwards
NASA Langley Research Center

Peter Eiseman
NASA Langley Research Center

Nabil M. Elhady
NASA Langley Research Center

Lars-Erik Eriksson
Aeronautical Research Institute (FFA)

S. Flaherty
NASA Langley Research Center

Neal T. Frink
NASA Langley Research Center

Peter Gettings
Aerodynamics Department
British Aerospace Aircraft Group

U. Ghia
University of Cincinnati

Peter Gnoffo
NASA Langley Research Center

Paul Gori
NASA Langley Research Center

T. R. Govindan
Penn State University

Randolph A. Graves, Jr.
NASA Headquarters

Lawrence Green
NASA Langley Research Center

William Gropp
Department of Computer Science
Stanford University

B. Grossman
Polytechnic Institute of New York

U. Gulcat
Georgia Institute of Technology

Clyde Gumbert
NASA Langley Research Center

Mohamed Hafez
NASA Langley Research Center

C. Hah
Penn State University

Billy Haigler
NASA Langley Research Center

Darryl W. Hall
Suite 1018
994 Old Eagle School Rd.
Wayne, PA 19087

John Hall
NASA Langley Research Center

H. H. Hamilton
NASA Langley Research Center

Andrew N. Harrington
School of Mathematics
Georgia Institute of Technology

Julius Harris
NASA Langley Research Center

A. Harten
NASA Langley Research Center

H. A. Hassan
Mechanical and Aerospace
Engineering Department
N. C. State University

Y. Hazony
Princeton University

Richard G. Hindman
ARO Inc./AEDC Div

John Hogge
NASA Langley Research Center

Ron Ho-Ni
Pratt & Whitney Aircraft Co.

Lona Howser
NASA Langley Research Center

James T. Howlett
NASA Langley Research Center

Gilbert H. Huffman
Penn State University

Larry Hunt
NASA Langley Research Center

Y. Hussaini
NASA Langley Research Center

James M. Hyman
Los Alamos Scientific Laboratories

Quyen Huynh
NASA Langley Research Center

Billy H. Johnson
U.S. Army Waterways Experiment Station

Gary M. Johnson
NASA Lewis Research Center

Jim J. Jones
NASA Langley Research Center

Wen-Huei Jou
Flow Industries, Inc.

Mary Anne Karczynski
NASA Langley Research Center

H. C. Kao
NASA Lewis Research Center

Edward Kansa
Lawrence Livermore Lab

Fred Kautz
CIA

G. David Kerlick
Nielsen Engineering & Research, Inc.

Scott Kjelgaard
NASA Langley Research Center

George J. Klem
U.S. Army Ballistic Research Lab

Doyle Knight
Department of Mechanical Engineering
Rutgers University

Edward Kowalski
Boeing Company

Gill Kramer
NASA Langley Research Center

Ajay Kumar
NASA Langley Research Center

Anil Kumar
Department of Engineering
Hampton Institute

Paul Kutler
NASA Ames Research Center

V. C. Laballeur
ONERA

V. C. Lai
U.S. Geological Survey

W. D. Lakin
Department of Mathematical Sciences
Old Dominion University

Jay Lambiotte
NASA Langley Research Center

Ki Dong Lee
Boeing Commercial Airplane Co.

Clark H. Lewis
Aerospace and Ocean Engineering
Virginia Polytechnic Institute
and State University

Eddie Liu
NASA Langley Research Center

Charles K. Lombard
PEDA Corp.

David Lovell
NASA Langley Research Center

Jim Luckring
NASA Langley Research Center

Levi Lustman
NASA Langley Research Center

Majeeb Malik
NASA Langley Research Center

Andrew Mark
U.S. Army Ballistic Research Lab

Fred Martin
NASA Johnson Space Center

Jim Martin
NASA Langley Research Center

C. Wayne Mastin
Mississippi State University

B. J. McCartin
Pratt and Whitney Aircraft

Geoffrey B. McFadden
251 Bleecker St. Apt 30C
New York, NY 10012

John E. Mercer
Flow Industries, Inc.

Marshall Merrian
NASA Ames Research Center

Jack Moran
5429 Wooddale Ave.
Edima, MN 55424

Gino Moretti
Polytechnic Institute of New York

Earl M. Murman
Department of Aeronautics
and Astronautics
Massachusetts Institute of Technology

Ejike Ndefo
Aerospace Corporation

Joseph P. Nenni
Aerodynamics Research Department
Calspan Corp.

Perry Newman
NASA Langley Research Center

Charles Nietubicz
U.S. Army Balistic Research Lab

David Nixon
Nielsen Engineering and Research Inc.

Youn H. Oh
Hughes Aircraft Co., Missiles
Systems Group

Samuel Ohring
David W. Taylor Naval Ship Research
and Development Center

Joseph Oliger
Department of Computer Science
Stanford University

Carl Edward Oliver
Air Force Office of Scientific Research

William J. Phares
Arnold Engineering Development Center

Zane Pinckney
NASA Langley Research Center

Barbara Pitts
NASA Langley Research Center

Joan Pitts
NASA Langley Research Center

M. M Rai
Department of Aerospace Engineering
Iowa State University

Allan W. Ratliff
2205 Lynn Rd.
Huntsville, AL 35810

Theodore A. Reyhner
Boeing Commercial Airplane Co.

W. Roberts
NASA Langley Research Center

David Roscoe
Scientific Research Association

M. E. Rose
NASA Langley Research Center

David Rudy
NASA Langley Research Center

Manuel Salas
NASA Langley Research Center

James A. Schmitt
U.S. Army Ballistic Research Lab

Floyd Shipman
NASA Langley Research Center

John Shoosmith
NASA Langley Research Center

Michael J. Siclari
Grumman Aerospace Co.

Bruce Simpson
Department of Computer Science
University of Waterloo

Bill Small
NASA Langley Research Center

Ray Smith
Engineering Analysis

Peter M. Sockol
NASA Lewis Research Center

Ricardo Solis
Naval Surface Weapons Center

Bharat Soni
Arnold Engineering Development Center

Reese L. Sorenson
NASA Ames Research Center

Jerry South
NASA Langley Research Center

Bob Smith
NASA Langley Research Center

Don Speray
NASA Langley Research Center

L. W. Spradley
Lockheed Missiles and Space Co.

K. P. Sridhar
Pratt & Whitney Aircraft

Joseph L. Steger
Suite 204
298 So. Sunnyvale
Sunnyvale, CA 94086

John Steinhoff
Grumman Aerospace Co.

Thomas Stephens
Beers Associates

Charlie Swanson
NASA Langley Research Center

Kuo-Yen Szema
Virginia Polytechnic Institute
and State University

Noel Talcott
NASA Langley Research Center

John Tanner
NASA Langley Research Center

Abraham Tassa
Lockheed Georgia Company

Frank B. Tatom
Engineering Analysis, Inc.

Kenneth E. Tatum
McDonnell Aircraft Company
McDonnell Douglas Corporation

Geoff Tennille
NASA Langley Research Center

Frank Thames
NASA Langley Research Center

Jim Thomas
NASA Langley Research Center

P. D. Thomas
Applied Mechanics
Lockheed Palo Alto Research Lab

Joe Thompson
Mississippi State University

Noel Thyson
AVCO Corporation

E. Turkel
NASA Langley Research Center

J. Th. Van Der Kolk
National Areospace Laboratory

B. Van Leer
NASA Langley Research Center

Marcel Vinokur
NASA Ames Research Center

Bob Voight
NASA Langley Research Center

Z. U. A. Warsi
Department of Aerospace Engineering
Mississippi State University

Beth Weidner
NASA Langley Research Center

Barbara Weigel
NASA Langley Research Center

Jim Weilmuenster
NASA Langley Research Center

Burton Wendroff
Los Alamos Scientific Lab

R. P. Weston
NASA Langley Research Center

John Whitcomb
NASA Langley Research Center

Woodrow Whitton
NASA Langley Research Center

George Widhopf
Aerospace Corporation

Irma Wilson
NASA Langley Research Center

James Wilson
Air Force Office of Scientific
Research

Y. S. Wong
NASA Langley Research Center

Richard Wood
NASA Langley Research Center

Steve Wornom
NASA Langley Research Center

James C. Wu
Georgia Institute of Technology

Steve Yaros
NASA Langley Research Center

Carson Yates
NASA Langley Research Center

A. T. Yen
303 University Club
Blacksburg, VA 24060

Shee-Mang Yen
Department of Aeronautical and
Astronautical Engineering
University of Illinois

Warren Young
NASA Langley Research Center

1. Report No. NASA CP-2166	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle NUMERICAL GRID GENERATION TECHNIQUES		5. Report Date January 1981
		6. Performing Organization Code 505-31-83-02
7. Author(s)		8. Performing Organization Report No. L-14195
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		10. Work Unit No.
		11. Contract or Grant No.
		13. Type of Report and Period Covered Conference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546 and Institute for Computer Applications in Science and Engineering Hampton, VA 23665		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The numerical generation of grids for the solution of partial differential equations is presented in a series of papers. Complex variable techniques, differential systems techniques, and algebraic techniques with applications in two and three dimensions are described. Adaptive grid generation and grid effects on errors in the solution of partial differential equations are additional subject areas that are discussed. The material in this document has been presented at the workshop on Numerical Grid Generation Techniques held at NASA Langley Research Center October 6-7, 1980.

17. Key Words (Suggested by Author(s)) Grid generation Coordinate transformations Partial differential equations Finite difference techniques Finite element techniques Fluid Flow solutions	18. Distribution Statement Unclassified - Unlimited Subject Category 34

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	574	A24

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1980

569

END

March 6, 1981